Enhancing Computational Methods for Strain Typing and Separating Strains of *Mycoplasma bovis* in Mixed Culture

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Matthew Waldner

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9 Canada

Or

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

Abstract

There are no programs that allow a user to isolate strain-specific sequences within a complex assembly of mixed bacterial strains, unbiased by reference assembly. The tools that do exist each have a specialized focus, such as isolating small haplotype differences within strains, or have a reliance on reference genomes that may bias the sequences. For this purpose we have developed a tool called the Separator of Strain Inherent Sequences (SepSIS) that extracts sequences specific to each bacterial strain from the *de novo* assembly graph created using the SPAdes assembler. SepSIS is accompanied by a set of pre-processing scripts that form the "SepSIS pipeline". The scripts are available at "`https://github.com/MatthewWaldner/sepsis`". The SepSIS pipeline provides two functionalities, with each accepting a particular form of input data. The pipeline was designed for use with Illumina MiSeq paired-read data, but in theory, any read dataset compatible with SPAdes could function with SepSIS. The first function of the SepSIS pipeline accepts reads obtained from non-clonal bacterial isolates as input. It then attempts to isolate the complete strain-specific sequences using relative coverage levels of strain-specific subsequences in the assembly graph. It is marginally successful at this task. The second function of the SepSIS pipeline accepts reads from independently cultured isolates and mixes them *in silico* before assembly. After assembly, the contiguous sequences are analyzed by SepSIS using meta-information describing their strain of origin to produce lists of sequences specific to each strain. These sequences can then be studied and contrasted further.

The second functionality of SepSIS was used to perform two primary investigations. The first investigation identifies unique sequences from sets of isolates, where each set was hypothesized to consist entirely of copies of a single strain. This investigation analyzed 10 sets of 5 independently sequenced isolates of *Mycoplasma bovis*, with all the isolates originating from a single culture spread on a growth plate. Despite originating from a single culture, it was found that many of the isolates had unique sequences; therefore, these isolates likely each represent an individual strain. The second investigation was based upon mixing two or more strains with contrasting phenotypic features allowing the second function of SepSIS to be applied to isolating sequences potentially responsible for each phenotype. By running multiple mixes with the same contrasting phenotypic combinations, the intersection of sequences common to a phenotype can be identified. This type of investigation was performed on 29 pairs of *Mycoplasma bovis* lung and stifle joint isolates, with each pair originating from a single animal. Infection location was considered a phenotype and sequences unique to each infection location were isolated and identified. The sequences with the strongest correlation to phenotype were variants of *Mycoplasma bovis* insertion sequences, or were from genes for variable surface lipoproteins and HAD-family hydrolases. The results show that SepSIS is useful when provided with reads sequenced from independently cultured isolates along with meta-information.

Acknowledgements

I would like to give special thanks to my supervisors, Tony Kusalik and Murray Jelinski. I would like to thank Tony Kusalik for his constant assistance and advice during my time in the Bioinformatics Lab at the University of Saskatchewan. I would like to thank Murray Jelinski for his support and for providing me learning opportunities in both the biological and computer science worlds throughout my thesis. Thank you to Karen Gesy for tutoring me in lab techniques, providing data, and for giving writing advice. I would also like to thank Andrea Kinnear for her help. Thank you to everyone in the clubroot research group at the U of S including Peta Bonham-Smith, Chris Todd, Yangdou Wei, Edel Lopez, and Solmaz Irani.

And most of all I would like to thank my parents for their unwavering support, encouragement, and assistance throughout my education.

Contents

List of Tables

List of Figures

List of Abbreviations

AMR	AntiMicrobial Restistance
ANI	Average Nucleic Identity
BAM	Binary Alignment Map
BP	Base Pair
BLAST	Basic Local Alignment Search Tool
BRD	Bovine Respiratory Disease
CIGAR	Concise Idiosyncratic Gapped Alignment Report
Contigs	Contiguous sequences
CSCC	Cyclic Strongly Connected Component
DDH	DNA-DNA Hybridization
E-Value	Expect Value
FASTA	Fast Alignment
GEE	Generalized Estimating Equations
GWAS	Genome-Wide Association Study
ISCC	Isolated Strongly Connected Component
LAPP	List of All Possible Paths
LOMP	List of Merged Paths
MLST	Multi-Locus Sequence Typing
MNPs	Multiple Nucleotide Polymorphisms
NCBI	National Center for Biotechnology Information
PPLO	PleuroPneumoniaLike Organism
PPV	Positive Predictive Value
SCC	Strongly Connected Component
SAM	Sequence Alignment/Map
SAVAGE	Strain Aware Viral Genome Assembly
SepSIS	Separator of Strain Inherent Subsequences
SPAdes	St. Petersburg genome Assembler
SNP	Single Nucleotide Polymorphism
SNV	Single Nucleotide Variant
VSP	Variable Surface Lipoprotein
WGS	Whole Genome Sequencing

1 Introduction

Bacterial species are currently divided into specific strains based on genotypic differences. These may range from single nucleotide differences to large scale DNA sequence differences. Most current sequencing technologies for bacterial DNA rely on DNA extracted from multi-cellular bacterial isolates grown in individual colonies. There is no guarantee that the bacterial colonies (isolates) are axenic (meaning that the isolate originates from one genotypically-distinct strain of a bacterial species with no other contaminating organisms). If a given isolate is not axenic (e.g. contains multiple strains) and a genome sequence of the isolate is assembled with any one of many existing assemblers, genotypic features distinct to a strain may be lost or obscured.

Tools do exist for the purpose of identifying genotypic variations in a sequence assembly. Each tool is tailored to specific tasks and production of output that is simple to parse and interpret. For example, the *de novo* genome assembler ALLPATHS focuses on reporting ambiguous DNA subsequences within contigs and EVORhA is a reference-based assembler that identifies haplotypes within an assembled whole genome [8, 33]. The issue for most of these tools is that they fail to represent complex and long variations (e.g., a novel location of an insertion sequence within a genome, or a gene that appears only within one strain of a species) of sequences within an assembly. A simplified example of such variations is given in Figure 1.1. However, there is an assembler that outputs an unsimplified assembly graph, described in Section 2.7, that can represent these complex variations of sequences. The *de novo* St. Petersburg genome assembler (SPAdes) produces such output, represented by a set of contiguous DNA sequences linked together in an assembly graph [5]. These individual contiguous DNA sequences within the assembly graph are referred to in this thesis as DNA subsequences. While the assembly graph allows multiple splits and joins between DNA sequences within the assembly, it does not describe which sequences within the output could be specific to a single strain in a mixture of strains.

Therefore, we have developed a tool to identify the strain-specific sequences within an assembly graph for a mixture of bacterial strains. This tool is named the Separator of Strain Inherent Sequences or SepSIS. SepSIS is accompanied by preprocessing scripts to assist with use of the tool. The tools and scripts make up the SepSIS pipeline and are available at "`https://github.com/MatthewWaldner/sepsis`". The algorithm in SepSIS iteratively parses the assembly graph produced by SPAdes in order to isolate sequences specific to particular strains within an assembly generated from a sequenced isolate containing multiple strains. These strain-specific sequences are output with subsequences that are common to all strains in the mix (strain-independent subsequences) on one or both ends. This is performed to allow for easier identification

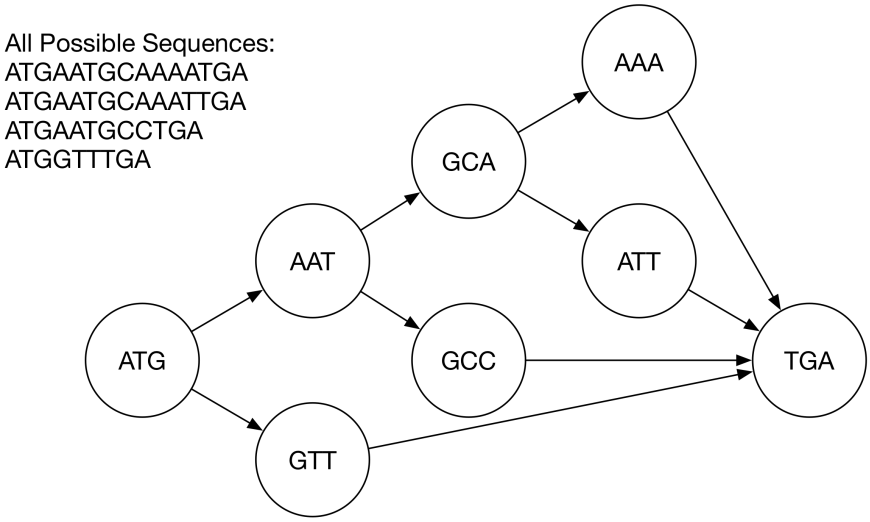

Figure 1.1: A graph where possible paths form four sequences. Subsequences are represented in the graph by the nodes, and all possible sequences created from the graph are displayed in the top left. This is a heavily simplified example of the sequences represented by a SPAdes assembly graph. For context, the individual sequences represented by SPAdes assembly graph are at least 55 nucleotides long and may range up to the thousands of nucleotides in length.

and positioning of the strain-specific sequences if compared to a reference genome. The combination of a strain-specific subsequence and one or two strain-independent subsequences is referred to as a strain-specific sequence. The specific criteria SepSIS uses to assign strain-specificity depends on the type of input from the user.

SepSIS has been developed with two primary functionalities, each with different input. The first function requires as input an assembly graph created using SPAdes and attempts to isolate and discern the sequences specific to non-clonal strains within it using relative coverage levels. Testing of this method was performed using assembly graphs from multiple *in silico* mixed sets of reads to simulate an isolate containing multiple strains, as well as isolates that had been mixed *in vitro* and then sequenced and assembled. This function of SepSIS produced only marginally successful results for the *in silico* mixed data and no positive results for the *in vitro* mixed isolates due to inconsistent read coverage in the assemblies. The second function takes as input an assembly graph created by running SPAdes with an *in silico* mixed set of reads, and a BAM file of the raw reads mapped to the assembly graph. The BAM file contains meta-information about the origin strain or read set of each read. SepSIS uses these files to identify the strain-specific subsequences and strain-independent subsequences.

Mycoplasma bovis is a pathogenic species of bacteria commonly found in cattle that can cause a variety of diseases. SepSIS was applied to datasets of previously sequenced *M. bovis* isolates along with newly grown and sequenced isolates. The data was provided through collaboration between the Department of Computer Science and the Western College of Veterinary Medicine. Two biological analyses, beyond general use of

SepSIS, were explored with these datasets using the second function of SepSIS discussed above.

The first biological analysis involved investigating the existence of non-clonal bacterial populations on a culture plate. Broth-grown *M. bovis* culture was streaked on 10 culture plates to allow for individual colonies of the bacteria to grow. From each culture plate, 5 individual colonies were then isolated and each isolate was theoretically axenic. These isolates were sequenced, and an analysis was conducted on the data using SepSIS. The analysis showed the presence of strain-specific sequences in the data from 16 out of the 50 colonies. Therefore, it was concluded that cultures on a single culture plate after growth can be non-clonal and exhibit genetic differences.

The second analysis was an investigation into the mechanism used by *M. bovis* to infect multiple anatomical locations. Included in an *M. bovis* dataset were reads from 29 pairs of *M. bovis* isolates, with each pair coming from the lung tissue and the stifle joint of a single animal. The data from these pairs were mixed *in silico* and run through SepSIS to produce strain-specific sequences for the lung and joint strains. These sequences were then pooled by phenotype (i.e. location of infection) and the pools compared. Common (across both pools) sequences were removed leaving phenotype-specific sequences. The genes containing these phenotype-specific sequences were identified. Most commonly, the phenotype-specific sequences mapped to insertion sequences, variable surface lipoproteins, and HAD-family hydrolases. There is supporting literature for these gene families to affect infection location, giving weight to these results [17, 42, 43, 48]. The most heavily supported result is a possible link between variable surface lipoproteins and tropism [42, 43]. Variable surface lipoproteins affect the binding of textitM. bovis to host cells, as well as modulate the response of that binding in host tissue. Research into these results will be pursued further post-thesis.

The remaining chapters are structured in the following manner: Chapter 2 contains background information on relevant topics, including the *M. bovis* species, bacterial strains, short read genome sequencing, graph theory, and relevant assembly programs and file types. The proposed research objectives are described in Chapter 3. Chapter 4 is an in-depth description of the materials and methods for the thesis, including a breakdown of the SepSIS algorithm; a description of the *M. bovis* datasets and how they were generated; pre-processing and post-processing steps taken on the data; and a description of the analysis performed on the post-processed data. The results from the analyses are presented and explained in Chapter 5. Chapter 6 contains a discussion of the development of SepSIS, the uses and shortcomings of the algorithm, the testing output of SepSIS, the results from an analysis of SepSIS, and how SepSIS relates to similar existing algorithms. Finally, Chapter 7 contains future work to be done on the *M. bovis* data and the conclusions of this thesis.

2 Background

2.1 *Mycoplasma bovis*

Mycoplasma bovis is a pathogenic intracellular bacterium that primarily infects cattle and bison [39]. This bacterium has several remarkable qualities, including the ability to survive for long periods of time in non-host environments. For example, it has shown the ability to survive in bedding sand for 8 months [18]. This is problematic for the prevention of spreading the bacteria due to the possibility of host infection with *M. bovis* through environmental contact, as well as contact with other infected animals. The cells are small, only 300 to 800 nm in diameter, and appear round and white when grown in culture. They also lack a cell wall, therefore the cells have no rigid form and are pleomorphic [26]. Infection with *M. bovis* is a significant cause of disease in both cattle and bison, resulting in bovine respiratory disease (BRD), arthritis, tenosynovitis, ear infection, abortion, and mastitis [39]. Despite the range of diseases caused by *M. bovis*, strains have also been isolated from the upper respiratory tracts of healthy cattle [26, 27]. It is notable that *M. bovis* has been found in other species as well, including whitetail deer and poultry [12, 29].

The spread of *M. bovis* is a cause for concern due to the negative economic impact. Estimates on the economic impact of *M. bovis* vary with study and location. One study showed that the net impact of BRD is a loss of $2,904,000,000 to the American beef cattle industry over 16 quarters [16]. The baterium was recently discovered in New Zealand and prompted an eradication effort. As of July 5, 2019 the New Zealand government has spent $234,000,000 on eradication and compensation pay-outs [31]. Therefore, the yearly cost of *M. bovis* when localized to large beef-producing country can be placed in the millions of dollars.

The genome of *M. bovis* has several unique characteristics such as having one of the smallest genomes of any bacterial species. The genome of the PG45 reference strain for *M. bovis* is the most cited and studied reference genome, possessing a length of 1,003,404 bases with 89% coding density [50]. Additionally, *M. bovis* has low GC content, with 29% of the genome of PG45 being G or C. This may cause bias during genome sequencing, as discussed in Section 2.6. Mycoplasma species uniquely decode the codon UGA as tryptophan instead of a STOP codon. Furthermore, this bacterium has limited metabolic pathways due in part to its small genome [7]. Therefore, it requires sugars, arginine, cholesterol and other sterols, peptides, and nucleotides for growth and reproduction [7].

2.2 Bacterial Species

The definition of a bacterial species has changed and evolved over time. The reason for this can be summed in the following quote: "The adequacy of characterization of a bacterium is a reflexion of time; it should be as full as modern techniques make possible. Unfortunately, one now regarded as adequate is likely, in 10 years time, to be hopelessly inadequate!" [10]. Originally, bacterial species were classified by microbiologists based solely on the phenotypic characteristics of a bacteria. Such criteria include cellular morphology and gram staining results [11]. The use of molecular biology approaches for describing bacteria has altered the concept of species from purely phenotypic descriptors to a combination of phenotypic and genotypic descriptors. More recent definitions of a bacterial species include detailed descriptions such as "a monophyletic and genomically coherent cluster of individual organisms that show a high degree of overall similarity in many independent characteristics, and is diagnosable by a discriminative phenotypic property" [40], as well as less precise definitions including "A species consists of strains of common origin which are more similar to each other than they are to any other strain." [11]. These definitions vary from source to source. However, when assigning a bacterium to a species from a working perspective, the common current methods use comparison of DNA sequences.

DNA-DNA Hybridization (DDH) used to be the primary method of species classification from a molecular biology approach, and still has descriptive value. DDH is a technique that measures the similarity between sets of nucleotide subsequences. The species definition for DDH states that 70% or greater DDH similarity between two genomes constitutes a single species [49]. This measure has been compared against a more modern genetic distance calculation of Average Nucleic Identity (ANI), as calculated by the Basic Local Alignment Search Tool (BLAST) [19]. This comparison used 70 closed genomes and found that for shared genes between strains an ANI of $\geq$ 94% corresponds to the 70% DDH similarity definition. However, the strains at the cutoff of 94% ANI were shown to differ in up to 35% of total genes. It was concluded that separating bacteria using a species-based definition failed to adequately describe intra-species genotypic and phenotypic diversity.

Currently, bacterial species is most often determined by the nucleotide sequence of the 16S rRNA gene. This technique originated, but was not yet commonly used, in 1985 following the description of ten major taxonomic groups of eubacteria through analysis of the gene's nucleotides [51]. The 16S rRNA possesses 9 regions with hyper-variable bases within its length, allowing for a high number of possible variations, and therefore possible classifications [15]. The usefulness of a small set of genes to predict genome relatedness was described by DR Zeigler in the paper "Gene sequences useful for predicting relatedness of whole genomes in bacteria" [54]. In this work, Zeigler describes that the similarity of certain encoding genes can accurately predict the relatedness of genomes. The current use of the 16S rRNA gene as a genetic descriptor for a bacterial species is proof of this concept. However, this method is less successful at describing variation within a species.

2.3 Bacterial Strains

Variation within the bounds defined by a bacterial species occur and are described in many different forms. Therefore, the precise definition of a bacterial strain is a contentious issue. Two of the most common definitions follow. A strain in the taxonomic sense as stated by Dijkshoorn *et al.* is "... made up of descendants of a single isolation in pure culture and is usually made up of a succession of cultures ultimately derived from an initial single colony" [11]. A second definition stated is of a strain in nature, described as "... an isolate or group of isolates that can be distinguished from other isolates of the same genus and species by phenotypic characteristics or genotypic characteristics or both" [11]. Under these definitions of a strain, the genotype of a strain may change over time and retain its identity. Unfortunately, this means that the term "strain" under these definitions can become less descriptive and precise over time.

In the same paper the author makes the point that a natural "strain" is rarely "pure". The practical issue is that unless the initial isolation of a bacterial species is monocellular, the isolation has the possibility of containing cells that have phenotypic or genotypic differences. In addition, genetic variation could originate from growth in the laboratory environment. For example, genetic variation within a sample has been reported in a paper analyzing the strains of *Bacillus anthracis* from the Amerithrax investigation. This investigation was performed into the *B. anthracis* spore samples sent in letters to 3 locations [35]. When cultured, these samples produced 4 morphological variations, each linked to a specific genetic alteration. One of several conclusions from this investigation was that a single sample could represent a mixture of these morphotypes, and therefore, strains [35].

For this thesis, a strain is defined as a member of a species that possess a wholly unique genotype. A single difference in a nucleotide between two genomes would mean that the two genomes belong to unique strains. This genotype change may or may not influence the observed phenotype. For example, a specific single nucleotide polymorphism (SNP) in the *M. bovis* genome is correlated with fluoroquinolone resistance. This SNP will cause the amino acid at position 83 in the *M. bovis* gene GyrA to mutate from serine to leucine or phenylalanine, both of which are linked to fluoroquinolone resistance [24]. This definition for strain currently is not an entirely practical ideal given that many methods for generating a genome sequence will not complete the genome, may generate errors within a genome sequence, or may mix multiple "strains" during the sequencing process. Despite this caveat, the stated definition will be used to increase the level of precision when discussing differences between genotypic "strains".

2.4 Genomics Definitions

This section consists of definitions and connotation for genomics terms used in this thesis. Contiguous sequences (contigs) are defined as the sequence constructed during assembly. A contig is not necessarily the longest possible product, referred to as a maximal product, of a set reads meaning that a contig may refer to

a partially assembled sequence of nucleotides or a subsequence of nucleotides within a larger contig. Scaffolds are linked contigs separated by gaps of known or estimated length. For the purposes of this thesis, both the terms sequence and subsequence, when used with reference to the contents of an *de novo* assembly graph (Section 2.7.2), are partially defined by the term contig. However, when the term sequence is used, it will refer to a maximal string of nucleotides. The term subsequence refers to a non-maximal string of nucleotides that, in context, is a part of a larger sequence. For example, if "ATCGATCGA" is a sequence, then some possible subsequences are "ATCG", "CGATC", and "GA". In the context of the assembly graphs introduced in Section 2.7.2, the term subsequence will most often be used to refer to the contig represented by either an incoming edge in an assembly graph or a node of an altered assembly graph (introduced in Section 2.7.2 and Section 4.3.4).

Subsequences may have multiple variants. These variants can be at the scale of single SNPs, multiple nucleotide polymorphisms (MNPs), a cluster of SNPs inherited together (a haplotype), or entire genes. Therefore, a strain-specific subsequence refers to a particular subsequence variant that is unique to a strain. Note that a strain-specific sequence may contain one or more strain specific subsequences. Suppose "TTAATTCCTT" and "TTGGTTAATT" are strain-specific sequences, then the strain-specific subsequences within would be "AA, CC" and "GG, AA" respectively. Note that all the "TT" subsequences are common between the two strains. Therefore, the "TT" subsequences are strain-independent subsequences and that a strain-specific sequence may contain strain-independent subsequences. Strain-specific subsequences may consist of only a few nucleotides, but they may also be thousands of nucleotides long. Strain-independent subsequences range from tens of nucleotides to ten-thousands of nucleotides.

2.5 Strain Identification and Assignment

2.5.1 MLST

Multilocus sequence typing (MLST) is a SNP-based method for categorizing strains of bacterial species [25]. An MLST scheme uses a predetermined set of SNPs located within a set of housekeeping genes for a species. Housekeeping genes are necessary for cellular function and expressed under all cellular conditions. This relates back to the definition of a bacteria strain near the end of in Section 2.2. Zeigler described how a set of encoding genes can be used to predict genome relatedness [54]. MLST functions using those principles to classify individual strains of a species.

A specified set of nucleotide variants present at the SNP loci within a single gene will allow assignment of that gene to a multilocus sequence type. As a whole, the set of multilocus sequence types form the descriptor for strain type. The primary method of strain determination in *M. bovis* is through Multilocus Sequence Typing (MLST). The standard scheme is described by Dr. Karen Register at the US Department of Agriculture [37]. The scheme for *M. bovis* uses 7 housekeeping (cellular maintenance) genes chosen from 6 *M. bovis* isolates, with each gene containing 4 to 7 SNPs. While there is a standard MLST for most bacterial

species, researchers may independently develop non-standard MLST schemes that include differing genes for specialized studies.

2.5.2 Other Computational Tools for Strain Identification and Assignment

There exist multiple methods for strain identification that vary in purpose. These methods vary based on the type of input sequences, raw reads or assembled genomes, and the level of precision to which the tool seeks to classify the sequence. There are 2 popular methods of strain identification: strain assignment through comparison to an existing database of reference strains, or strain identification through haplotype extraction.

Tools that function by assigning strains through sequence comparison with a database are often used to classify metagenomic read sets, but also accomodate fully-assembled genomes. For example, Kraken 2 is a metagenomic taxonomic classifier that takes as input a set of nucleotide sequences [52]. It then assigns each sequence a taxonomic label through comparison to an existing database of sequences annotated with species and strain information. StrainSeeker is another popular tool that takes as input a set of raw reads and searches a tree-based database of bacterial genomes for existing matches [38]. While Strainseeker provides a default tree containing a limited set of existing bacterial strains, the strength of the software comes from allowing the user to create their own guide tree from existing strains. However, these tools do not identify novel strain features; rather, they only assign sequences labels based on similar, previously sequenced strains.

Tools that specialize in extracting haplotypes require raw read sets and may require a reference genome. For the context of this thesis, a haplotype is defined as a strain with a particular set of SNPs present in the genome. Evolutionary Reconstruction of Haplotypes (EVORhA) takes as input a set of raw reads and aligns them against a reference genome. The existing bacterial haplotypes are identified based on SNP presence and frequency [33]. The software package Strain Aware Viral Genome Assembly (SAVAGE) was created to extract multiple viral haplotypes from within a set of reads without a reference genome [4]. These two pieces of software may not assign taxonomic labels, but instead they highlight and identify differences between strains within a read set.

2.6 Short-Read Whole Genome Sequencing

Whole genome sequencing (WGS) is the process of converting the whole genome of an organism to digital data. Short-read WGS is a subtype of WGS that refers to the length of the DNA fragments being sequenced. The exact length depends on the sequencing platform used. Illumina MiSeq is a commonly used short-read sequencing platform. The sequencing process performed by MiSeq during paired-end sequencing is shown in Figure 2.1. First, the DNA to be sequenced is fragmented to a prefixed base pair (bp) length, generally 300 bp or less and denatured. The forward and reverse ends of each DNA sequence, referred as 5' and 3', are ligated to adapter DNA sequences. One adapter then binds to a lawn of oligonucleotides on a sequencing chip. A polymerase then copies these DNA sequences and these first copies are subsequently washed away. The DNA

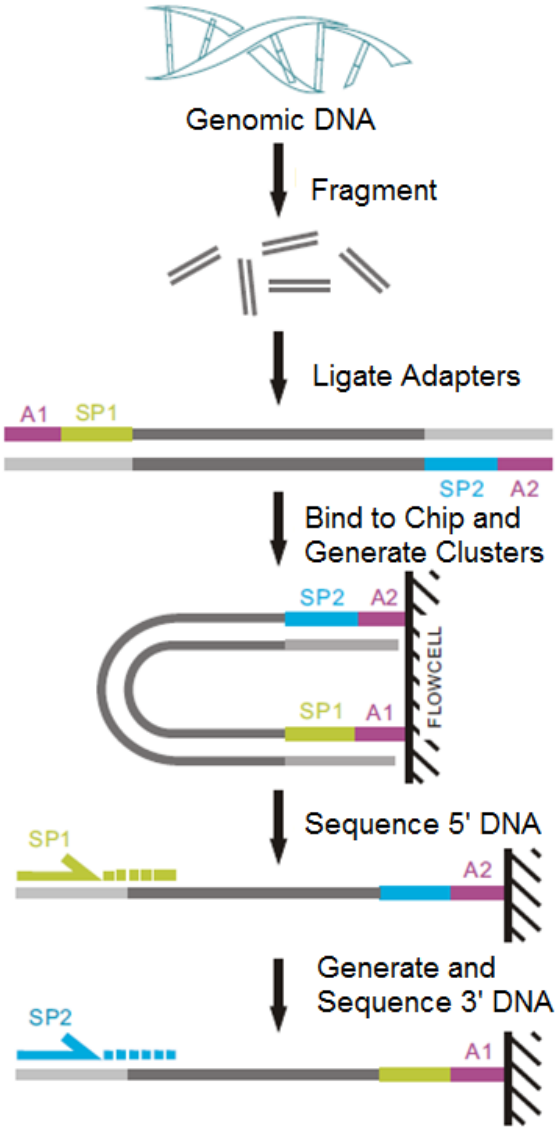

Figure 2.1: A summary of the steps taken during MiSeq short-read sequencing. Adapted from BGI's Whole-genome Re-sequencing [13].

sequence is then amplified repeatedly by binding both ends to nearby chip affixed adapters, copied with a polymerase (called bridge PCR), and denatured leaving only one end of each denatured sequence bound to the chip. This results in a cluster of identical nucleotide fragments. The 3' starting sequences are then washed off the chip, leaving only 5' sequences. Fluorescently-tagged nucleotides are added and bound to the forward sequences sequentially. As each nucleotide binds, a unique light signal is emitted. The cluster of adjacent nucleotides ideally provides a signal that is intense enough to be picked up by a detector. Alternatively, if the signal is weak due to improper nucleotide binding or a small nucleotide cluster, the signal detected may be wrong or of low quality. Afterwards, only the signal nucleotides are washed off and the 3' end of the DNA binds to the chip. The DNA strand is copied, denatured, and the 5' sequences are washed away. Again the signal nucleotides are added, producing the reverse sequences that are read by the detector. The resultant digital data is a file containing both forward and reverse reads with multiple reads of each fragment of DNA [14].

As a sequencing platform, MiSeq is a reasonably priced option that produces high quality reads with a low error rate of 0.8%. However, MiSeq has been shown to produce lower quality results during sequencing in highly repetitive regions [34]. Reads sequenced from highly repetitive regions are difficult or impossible to align due to repetitious nucleotides causing misplacement or incorrect nucleotide overlaps in an assembly [34]. Additionally, the assistance of a reference genome to align the reads against is only marginally useful [34]. MiSeq is also vulnerable to GC bias. GC bias refers to the bias of sequencing techniques towards genomes with intermediate GC content. Reads with high or low GC content are more vulnerable to sequencing errors resulting in their underrepresentation in the final sequencing product [2]. Despite these weaknesses, MiSeq is still a reliable and extensively-used WGS technology.

2.7 Graph Theory

2.7.1 Graphs and Cycles

Graphs are used to model pairwise relationships between a set of objects and are constructed using two elements: nodes and edges. Nodes represent objects, information, or ideas, while edges describe relationships between the nodes. The edges in a graph may be directed or undirected in their description of a relationship. A directed edge describes a one-way relationship between nodes. A bi-edge (also known as an extraverted edge) is a single edge that describes a two-way relationship between nodes. An undirected graph has no directed edges, a directed graph has directed edges, and a bidirected graph can have multiple edge types, including bi-edges that have directed relationships in two directions, and will not be discussed further in this section.

Figure 2.2 shows an undirected graph, and Figure 2.3 shows a directed graph. Both graphs have nodes labeled by letters and edges labeled by integers. A path within a graph is a series of non-overlapping nodes and edges. For example, in Figures 2.2 and 2.3, one path would be through nodes ABCF. A cycle is a series

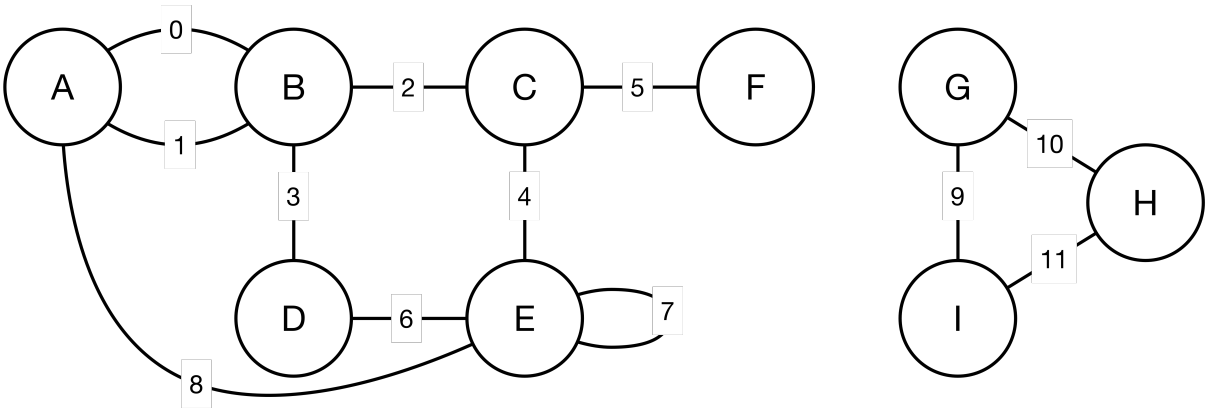

Figure 2.2: An example of a graph with undirected edges. Nodes are labeled with letters while edges are labeled with numbers.

of non-overlapping nodes and edges, except that the first and final nodes must overlap. In Figures 2.2 and 2.3, a cycle exists through nodes ABCEA. Components are defined as the maximal subset of an undirected graph where each node in the subset is connected to every other node in the subset by a path. There are 2 components in Figure 2.2, consisting of nodes ABCDEF, and GHI. Branch nodes have multiple other nodes preceding or succeeding them in a graph. In Figure 2.4, nodes A, C, E, G, H, L, and N are the branch nodes. Terminal nodes are nodes with either no successor or no predecessor nodes. In Figure 2.4, nodes M and O are terminal nodes.

Strongly connected components (SCCs) are maximal components where every node within the component is reachable from every other node. In Figure 2.3, the nodes ABCDE make up an SCC. Node F is not included in that SCC because no other nodes can be reached from Node F. Note that a single node can be a SCC if it is not within another SCC. These single node SCCs are designated as Isolated Strongly Connected Components (ISCC) for this thesis, while components that consist of greater than 1 node are defined as Cyclic Strongly Connected Components (CSCC). This is because every node within an CSCC is within a possible cycle. Figure 2.4 shows a CSCC with nodes and edges orange, and multiple ISCC nodes in blue.

2.7.2 De Brujin Graphs and SPAdes Assembly Graphs

A de Brujin graph is a type of a directed graph. De Brujin graphs are extensively used in sequence alignment and assembly programs because of their ability to represent permutations of n-mers. An n-mer is a string of characters of length n. The first 1 to n-1 characters of an n-mer are the n-mer prefix, while the last 2 to n characters of an n-mer are the suffix. For example, the word "READ" is a 4-mer, containing a prefix of "REA" and a suffix of "EAD". In a de Brujin graph, an edge contains the full n-mer formed by the prefix and suffix. Each node contains n-1 characters that represent the prefix for outgoing edges and the suffix for incoming edges. Figure 2.5, shows a de Brujin graph of the strings "`GATGCAAT`" and "`GATGGAAT`" made of 4-mers. A node is designated as a branch node, coloured in grey, when it has more than 1 incoming or

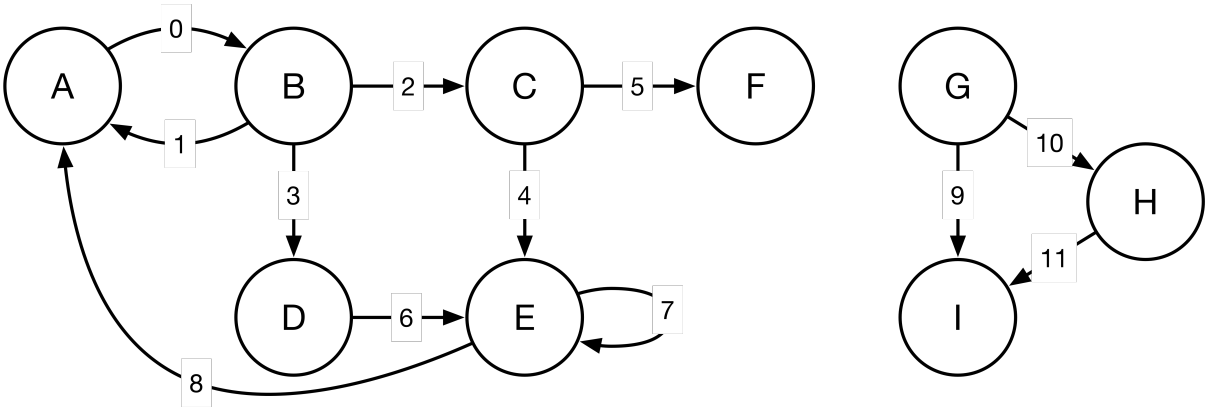

Figure 2.3: An example of a directed graph. Nodes are labeled with letters while edges are labeled with numbers.

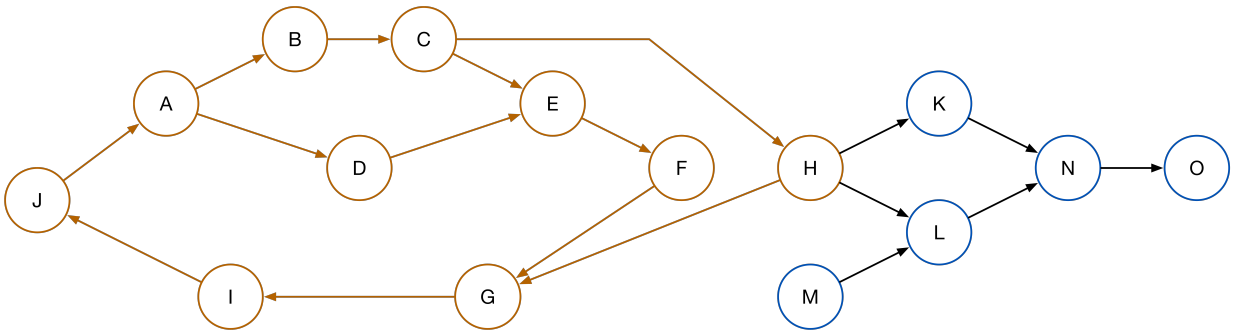

Figure 2.4: A graph containing one cyclic strongly connected component (CSCC) and five isolated strongly connected components (ISCC). Note that each ISCC does not contain any edges.

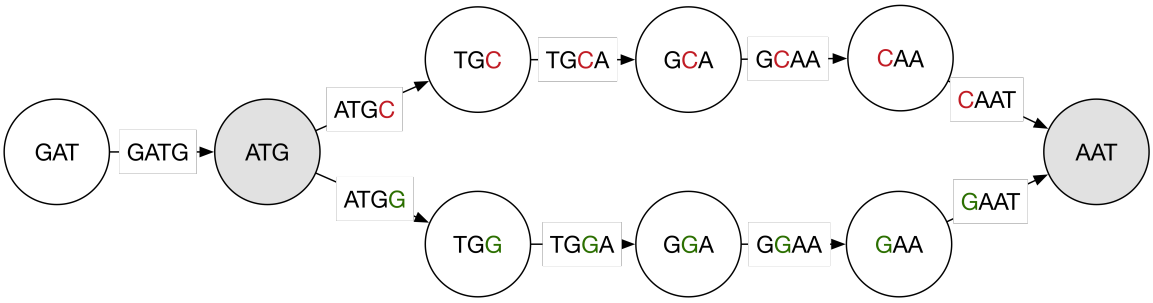

Figure 2.5: An example of a de Brujin graph.

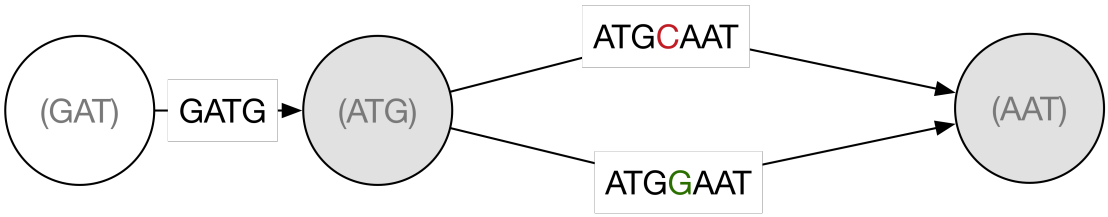

Figure 2.6: An example of an assembly graph. This graph contains the same data as Figure 2.5, but in assembly graph form. The value of k for the graph is set to 3. Note that the nodes do not actually contain or represent the characters shown. They exist as an example of the minimum length 3-mer prefixes and suffixes.

outgoing edges.

An assembly graph, as created by the SPAdes assembler described in Section 2.10.1, is a compressed form of a de Brujin graph. In a SPAdes assembly graph, edges contain all character strings and nodes merely exist as a form of connection between edges. An example of an assembly graph is given in Figure 2.6, and is an alternate form of the de Brujin graph in Figure 2.5. Assembly graphs contain prefixes and suffixes of overlapping characters, similar to de Brujin graphs. In assembly graphs the prefixes and suffixes have a minimum length of k, where k is a specified k-mer size given to SPAdes during assembly. However, irrespective of the value of k, there is no set length for a subsequence contained by an edge. Therefore, assembly graphs can represent the same information present in a de Brujin graph to be represented in a compressed manner.

In a true assembly graph the subsequences are contained in a single incoming edge. However, in Chapter 4, the term node is referred to as containing subsequences. This is because the assembly graph's structure is altered to make calling subsequences simpler by placing all of the subsequences (and other relevant information) within nodes and linking them with edges. In Section 4.3.4 an example is given of this graph structure alteration.

2.8 Genome Assembly

Genome assembly is the aligning and merging of subsequences to reassemble a larger whole sequence. Genome assembly can be performed using a *de novo* or reference assembler. A reference assembler takes as input a completed genome of the organism to act as an alignment template for the assembly process. Reference assembly allows for genome assembly under non-optimal circumstances, such as if a set of reads does not cover the entire genome, or when a set of low quality reads are encountered in a segment of the genome. In the case that stretches of the genome are missing from the read set, the reference genome may act as a template to fill in the gaps created by a lack of sequence data. However, reference assembly heavily biases the assembled scaffolds to the reference sequence, potentially obscuring features unique to the strain undergoing assembly.

A *de novo* assembler does not require a completed reference genome, instead the assembly is created by iteratively overlapping the input reads to form contigs. While this prevents reference bias, *de novo* assembly may lead to misassembly of the genome. This can be caused by the assembler algorithm, by low quality reads containing gaps in the full genome, or highly repetitive subsequences.

The result of either assembly method will produce a variety of output files. There will always be one type of file containing an assembled sequence or sequences, such as a SAM, BAM, FASTA, FASTQ, or FASTG, all discussed below in Section 2.9. Output may also include the coverage of the sequences and a measure of quality for the bases or subsequences within the assembled sequences. A variety of algorithms exist for both types of assembly, with different algorithms tailored to specific organisms and read types. Two of these algorithms are described in more detail in Section 2.10.

2.9 Relevant File Formats

2.9.1 FASTA Format

The fast alignment (FASTA) format is a text-based format for representing nucleotide or peptide sequences in strings. Each represented sequence within a FASTA file consists of two sections. The first section is the sequence header, represented on its own line in the text file. The sequence header always starts with a ">" character, and contains contextual information on the sequence. The FASTA files used in this thesis output by SPAdes, discussed in Section 2.10.1, make use of 6 of the 17 accepted nucleic acids symbols. A,C,T, and G represent the nucleic acids adenine, cytosine, thymine and guanine. N represents one ambiguous nucleotide (any one of the previous four nucleic acids), and the character "-" represents a gap in a sequence alignment. Newline characters within a FASTA file are removed or ignored by most software making sequences divided into multiple lines valid as input. Below are the header and first 140 characters of the PG45 *M. bovis* reference genome in FASTA format:

```
>CP002188.1 Mycoplasma bovis PG45 clone MU clone A2, complete genome
```

```
ATTTATATATGAATATCAATAGCACTAATGATAAGGAAATTGCTTTAAAGTCTTACACTGAAACCTTTTTAGATATTCTGAGACAAGAATT
AGGCGATCAGATGCTTTATAAAAACTTTTTTGCAAATTTTGAAATCAAA
```

2.9.2 FASTQ

The FASTQ format is an expanded version of the FASTA format with additional information on the quality of each nucleotide. The FASTQ format expresses this information in FASTA format for the first 2 lines of each sequence, with a "@" at the beginning of the header instead of a ">". The FASTQ file also contains a third line with a "+" character indicating the separation between the nucleotide sequence and the 4th line, and a final 4th line containing a string of characters. Each character in the 4th line indicates a quality score for each base in the nucleotide sequence. In the Sanger quality score format, the quality score of the nucleotide + 33 is equivalent to the character's ASCII code, with quality scores from 0 to 40. Like FASTA format, newline characters in the 2nd and 4th lines are removed or ignored by most software.

Below are the 140 characters the PG45 *M. bovis* reference genome in FASTQ format with a mockup quality score line:

```
@CP002188.1 Mycoplasma bovis PG45 clone MU clone A2, complete genome
ATTTATATATGAATATCAATAGCACTAATGATAAGGAAATTGCTTTAAAGTCTTACACTGAAACCTTTTTAGATATTCTGAGACAAGAATT
AGGCGATCAGATGCTTTATAAAAACTTTTTTGCAAATTTTGAAATCAAA
+
1BDDA0B33F0AB1113F122D1DEFH2100BBA1BF11BBG2AF1AAD1D1BAF1DF0GG/1E11F11A112B12D12DF2F22BBGH//
F2BF22BD211BB2BB22FG111B1B1111112BE221BDG00F0110F
```

2.9.3 FASTG

The FASTG format SPAdes produces is similar to the FASTA format, but includes additional information about the structure and elements of the assembly graph. A subsequence header contains information about a single incoming edge containing a single nucleotide sequence, and multiple outgoing edges. This is represented in the form >incoming:outgoing,outgoing,etc. In the example below, the first two lines have been wrapped, and represent the single header line of the file. The information for each edge includes an edge id, the length of the subsequence in the edge, and the coverage of the sequence as calculated by SPAdes during assembly. In the example below, the incoming edge has an ID of 9692, a length of 122, and a coverage of 97. The orientation of the nucleotide sequence is also indicated by a symbol. If a " ' " follows the edge information, the nucleotides represented by that edge are in the 3' - 5' orientation. If the symbol is not present, the nucleotides are in the 5' - 3' orientation. Edge 631054 below is in the 3' - 5' orientation, and the other example edges are in the 5' to 3' orientation. Otherwise, the nucleotides are 5' to 3'. Unlike FASTA, there are no gaps or undefined nucleotides in the subsequence. The example of the format follows; note that the nucleotide subsequence had to be shortened for size constraints:

```
>EDGE_9692_length_122_cov_97.000000:EDGE_630574_length_240_cov_20.983193,
```

```
EDGE_631054_length_240_cov_75.369748',EDGE_94170_length_240_cov_4.487395;
ATTTATATATGAATATCAATAGCACTAATGATAAGGAAATTGCTTTAAAGTCTTACACTGAAACCTTTTTAGATATTCTGAGACAAGAATT
AGGCGATCAGATGCTTTATAAAAACTTTTTTGCAAATTTTGAAATCAAA
```

2.9.4 SAM/BAM

The Sequence Alignment/Map (SAM) format contains information on the alignment of query subsequences against a reference sequence. The Binary Alignment Map (BAM) format is the binary conversion of the SAM format. SAM files have 11 mandatory fields listed below. This format may also include optional fields with additional information about the alignment or sequence.

1. The name of the query sequence. In an alignment of reads against a reference genome, this would contain the name of a read.
2. A combination of bitwise flags describing the sequence, including information on alignment quality, secondary and supplementary alignments, reverse compliments, and other details.
3. The name of the reference sequence.
4. The first position of the sequence mapping to a reference base.
5. The mapping quality, represented by an integer.
6. The Concise Idiosyncratic Gapped Alignment Report (CIGAR) format string describing the matches, mismatches, insertions, deletions, and gaps in the alignment.
7. The name of the sequence pair or next sequence in the alignment.
8. The leftmost mapping position of the sequence pair or next sequence in the alignment.
9. The length of the template sequence against which the query sequence is mapped.
10. The nucleotide string of the query sequence.
11. An ASCII string of coded base qualities plus 33, identical in format to the quality string in FASTQ format.

2.10 Genome Assembly and Alignment Software

2.10.1 SPAdes *de novo* Assembler

SPAdes is a *de novo* genome assembly toolkit and is specialized for the assembly of short bacterial genomes [5]. SPAdes takes as input a set of reads and several run options. Most run options are irrelevant for the purposes of this thesis, except for the -k option. This option dictates the size of the overlapping k-mers used

during construction of assembly graphs and de Brujin graphs during assembly. SPAdes will output both the assembly graph from step 3 in FASTG format and the contigs in FASTA format. The SPAdes algorithm has four stages listed below.

1. An initial assembly graph is roughly constructed using k-mers from paired reads and error correction methods.

2. Bi-edges are created based upon the original assembly graph, creating a de Brujin graph.

3. An assembly graph is constructed through the simplification of the de Brujin graph.

4. The list of contigs are created through the simplification of the assembly graph. This step is performed through iteratively condensing and deleting the edges in the paired de Brujin graph containing overlapping subsequences, leading eventually to single edges containing contigs. Note that this step leads to information loss.

2.10.2 Minimap2

Minimap2 is a sequence alignment program that can function as a reference assembler for reads generated by the Illumina MiSeq platform [22]. Minimap2 takes as input a reference sequence and a set of DNA sequences to map against the reference sequence. To perform this, minimap2 employs what is known as a "seed-chain-align" methodology for sequence assembly. The "seed" step is the process of finding the minimum set of k-mers within the set of query sequences, generally a set of reads, to represent all k-mers within an assembly. This minimum set of k-mers is labeled as a set of "seeds". The seeds that are exact matches to the reference genome are labeled as "anchors" and overlapping anchors are matched together in "chains". Base-level alignment then extends the ends of the chains, creating the full alignment. Minimap2 will output the alignment in the SAM format discussed above in Section 2.9.4. The authors of minimap2 performed a comparison with Bowtie2 [20], BWA-MEM [21] and SNAP [53] showed that minimap2 is one of the fastest and most accurate genome assemblers available [22].

2.10.3 BLAST

The Basic Local Alignment Search Tool (BLAST) was created in 1990 by the National Center for Biotechnology Information (NCBI) as a rapid local sequence comparison tool [3]. The algorithm has been supported and updated over time by NCBI, with the currently supported version of the algorithm named BLAST+ [9]. BLAST+ is a local sequence alignment algorithm that aligns one or more query nucleotide or amino acid sequences against an existing reference database for similar sequences. There are existing target databases available, such as the NCBI RefSeq database [32]. In addition, a reference database can be created using a multi-sequence FASTA file. The BLASTN suite of BLAST+ specifically enables nucleotide queries against

nucleotide databases. Multiple output fields and formats exist for BLASTN. The default tab-separated output is designated format 6. The output fields are described below.

1. The name of the query sequence.
2. The name of the subject or target sequence.
3. The percentage of identical matches between the query and target sequences.
4. The length of the alignment between the two sequences.
5. The number of gap openings in the alignment.
6. The start position of the alignment in the query sequence.
7. The end position of the alignment in the query sequence.
8. The start position of the alignment in the subject sequence.
9. The end position of the alignment in the subject sequence.
10. The expect value (E-value) for the alignment. The E-value represents the expected number of alignments of similar quality that could occur by chance in the given database.
11. The bit score for the alignment. The bit score accounts for the size of a sequence database in which the alignment could occur by chance. The bit score is log2 scaled and normalized from the raw score with respect to the scoring system.

3 Research Goals

Multiple procedures exist to identify strains of bacterial species, with each method specializing in a particular form of data. Notable among these methods are MLST, whole genome comparison allowing strain assignment, and haplotype identification. MLST characterizes bacterial strains using a selected set of genes and SNPs, with these genes existing within all strains of a species [25]. MLST is useful as a generalized characterization system; however, a set of SNPs limited to a subset of genes prevent any expanded phenotypic study from being conducted. Strain assignment through whole genome comparison is useful for generalized identification of strains present within a sample [38, 52]. Haplotype identification tools identify differences in strains down to the SNP level across the whole genome and create either whole genome sequences containing variants, or files containing only the variants. This allows for studies of the differences between these haplotypes [4, 33].

Despite the range of tools available, there is currently an unfilled niche for tools that identify subsequences unique to particular strains (strain-specific subsequences) that range in size from a single SNP to genes that are thousands of nucleotides long in a *de novo* genome assembly. *De novo* assembly is useful for identifying strain-specific sequences because it is free of the possibility of altering strain-specific sequences through reference bias. Therefore, the goals of this thesis are derived from a *de novo* assembly approach to strain-specific sequence identification. The primary goals of this thesis are to:

1. Create a tool to isolate and identify the sequences unique to particular strains of *M. bovis* through analysis of intermediate information (assembly graphs) created during the *de novo* genome assembly operation. This tool is called the Separator of Strain Inherent Sequences or SepSIS.

2. Develop two datasets to use specifically for the development of, and experimentation with, SepSIS. One dataset is created through growing and mixing cultures *in vitro* and subsequent sequencing. The other dataset is created by *in silico* mixing of individually grown and sequenced isolates of *M. bovis*.

3. Develop an approach to verify the sequences produced as output from SepSIS as strain-specific.

4. Formulate answers to relevant biological questions using the developed method and data.

A further breakdown and expansion of the goals for this thesis follows. Each subsection *3.X* below corresponds to goal X above. In addition, a list of assumptions, non-goals, and study limitations are provided for further clarity.

3.1 Goals for the Creation of SepSIS

1. Design a coverage-based approach to identify and extract the sequences specific to particular strains of bacteria from a genome assembly. The approach will begin with a set of reads containing more than 1 strain of a single bacterial species, and then use relative coverage ratios between subsequences to identify which subsequences likely belong to each strain. In theory, if two strains of a bacterial species are mixed together, then the subsequences of the genome specific to each strain will have half as much sequencing coverage as the rest of the genome. These subsequences can then be extracted and placed in series if they are adjacent in the assembly graph. Producing these strain-specific sequences allows for further study using other tools and methods and is the purpose of SepSIS.

2. Design a verification method for SepSIS; that is, determine evidence that supports a conclusion that the strain-specific sequences are unique to, and characteristic of, a particular strain. The best ways to confirm the existence of strain-specific sequences is to complete a high coverage genome for that strain, or through wet lab PCR-based verification. However, these methods are unrealistic for the scope of this thesis. Therefore, the verification method will be derived from meta-data. The data will be obtained through the creation of a synthetic dataset, as discussed in Section 3.2.

3. Customize SepSIS to compensate for the highly variable coverage found in short-read *M. bovis* data. Mycoplasma species are notoriously difficult to culture due to highly specific growth environments [30]. The combination of this with regular sequencing anomalies such as GC bias can lead to highly variable coverage across a single genome. Therefore, SepSIS is specifically designed to handle poor and variable coverages within a read set.

4. Include strain-independent subsequences (subsequences shared among the strains in a mixture) on the end or ends of each of strain-specific sequence. There are multiple benefits to ensuring the strain-specific sequences output by SepSIS have terminal strain-independent subsequences. First, the strain-independent subsequences can be validated against existing reference genomes to ensure they are truly strain-independent and not misassembled. Second, it allows for the positioning of the strain-specific sequences when compared against a reference assembled genome. Third, it allows for primer design enabling wet lab investigation into the sequences.

3.2 Goals for Dataset Development

1. Generate a dataset from laboratory-grown cultures of *M. bovis*. This thesis is a collaboration between the Department of Computer Science and the Western College of Veterinary Medicine at the University of Saskatchewan. In addition to the computational element of the thesis, growth of the samples used in the thesis is also necessary. Because of the research focus on *M. bovis* in Dr. Murray Jelinski's lab, *M.*

bovis was chosen as the model organism for the development of SepSIS. This provided an opportunity for my participation in the wet lab work in order to obtain laboratory knowledge and context behind the data produced, and to better familiarize myself with *M. bovis*. The dataset produced contains both normally sequenced *M. bovis* isolates, as well as isolates that were intentionally mixed *in vitro* and subsequently sequenced.

2. Generate a synthetic dataset for the development of SepSIS. Because it is expensive to grow and sequence *in vitro* mixed isolates of *M. bovis*, additional mixes are synthetically created in order to help test SepSIS. These *in silico* mixes are made of the individual short-read sequenced *M. bovis* isolates grown in Dr. Jelinksi's lab. Additionally, the read sets from *in silico* mixes have meta-information on the origin isolate of each read. This provenance meta-information is not present for the isolate mixes generated *in vitro*. This meta-information serves as the basis for the verification method.

3.3 Goals for the Verification of SepSIS Output

1. Verify the results from the *in silico* and *in vitro* mixes run through the coverage-based method (3.1.1) against the results from the same mixes run through the verification method (3.1.2). Given that the verification method produces true strain-specific sequences, these results can be used as ground truth to evaluate the coverage-based method results. If the coverage-based method produces the same strain-specific sequences as the verification method, then it can be concluded that the coverage-based method succeeded. However, meta-data for verification only exists for the *in silico* mixed read sets. Therefore, the results from the *in vitro* isolate mixes must be compared against the results from the *in silico* isolate mixes consisting of the same isolates.

2. Examine how well SepSIS is able to handle larger mixes. SepSIS is designed to handle mixes of 2 or 3 isolates, but the ability to handle mixes of size 4 or 5 is examined as well. This is performed by comparing the output sequences of smaller mixes against the output sequences of larger mixes, with all mixes within a comparison drawing from the same bank of samples. The output sequences from cross-comparisons can then be examined to determine if larger mixes produce erroneous strain-specific sequences.

3.4 Goals for Experimentation Using SepSIS

1. Determine if multiple genomically distinct strains of *M. bovis* exist on a single colony growth plate at the same time. Part of the *M. bovis* dataset from Section 3.2.1 consists of sequencing data from multiple sets of isolated colonies from a single growth plate. By contrasting the SepSIS output of varying *in silico* mixes created from isolates belonging to a single growth plate, the presence of strains with unique sequences can be confirmed.

2. Explore the effects of contamination with different species on the SepSIS pipeline. This is performed by creating *in silico* mixes of *M. bovis* isolates and isolates of other *Mycoplasma* species. The sequences produced from the SepSIS runs on the mixes are examined to determine if sequences belonging to non-*M. bovis* appear in the SepSIS pipeline output after filtering steps designed to remove them.

3. Identify sequences specific to *M. bovis* isolates with tissue tropisms in the stifle joint and the lungs of cattle. The dataset contains multiple pairs of *M. bovis* samples from both the stifle joint and lung of a single bovine (3.2.1). Currently, the particulars of why certain strains of *M. bovis* infect a particular tropism are unknown. A list of sequences potentially affecting tropism is produced by creating *in silico* mixes of the two isolates originating from different tissue tropisms from a single animal and extracting the strain-specific sequences using SepSIS. These tropism-unique sequences are then compared with tropism-unique sequences from other isolates with the same tropism to find sequences common to that tropism.

3.5 Assumptions, Non-Goals and Limitations

1. The goal of this thesis is not to create an assembler, or to output fully assembled strain-specific genomes. The primary goal is to determine if coverage-based comparisons can be used to extract strain-specific sequences for further study. This limitation is due to the variable coverage in the dataset used.

2. A general assumption made in microbiology laboratory work is that each isolated and sequenced colony of *M. bovis* is clonal at the point of isolation. However, it is possible that an isolated colony contains non-clonal *M. bovis* strains. If this is the case, the read set resulting from the sequencing of this colony will unknowingly represent more than one strain. Despite this, we assume that for the synthetically (*in silico*) mixed read sets, each individual read set within a mix truly only consists of one strain.

3. There are no strict goals concerning runtime or memory usage performance. However, performance requirements suitable to allow a batch of isolate mixes to run on a modern server-grade computer in minutes or hours (rather than days or weeks) are necessary for practical use of the algorithm.

4. There is no goal to provide a graphical user interface. SepSIS consists of a series of scripts executed from a command-line interface. However, the primary SepSIS scripts are able to be used by bioinformaticians other than the author without difficulty.

5. This thesis is limited to *M. bovis* data, sequenced with an Illumina Miseq platform producing paired short-reads. However, SepSIS is likely capable of functioning using data from other platforms that also function with the SPAdes assembler.

6. SepSIS is limited by the capabilities of other software used in the pipeline. The pipeline will make use of publicly available software, namely sequence assemblers, sequence aligners, and python packages.

SepSIS inherits any flaws that software may contain.

4 Data and Methodology

The data and methodology chapter is split into 5 sections. Section 4.1 contains both a high-level introduction to the collection of preprocessing scripts needed for the SepSIS tool and SepSIS itself. Together these make up the SepSIS pipeline. A detailed mid-level description of the various steps taken in the SepSIS pipeline is also present in this section. This mid-level description acts as a moderately detailed summary of all steps in the methodology chapter. Section 4.1 also contains an example of an assembly graph in Figure 4.1, and an abstract workflow of the preprocessing steps leading up to the use of the SepSIS algorithm in Figure 4.2. Section 4.2 describes in-depth the preprocessing steps necessary for the use of SepSIS. Figure 4.3 is a workflow diagram that encompasses the preprocessing steps described in the section, as well as their relation to the coverage-based method (separated into two coverage-based modes) and the verification method of SepSIS. Section 4.3 describes the whole of SepSIS, which encompasses all the sub-algorithms within and the differences between the coverage-based modes and differences between the coverage-based method and verification method. Figure 4.4 describes the input parameters and the internal algorithmic steps of the SepSIS. The verification method's algorithm involves changes to several functions used in the coverage-based method meant to evaluate adjacent nodes in the assembly graph. Therefore, it is not represented in Figure 4.4, but is discussed in Section 4.3.6. Section 4.4 contains a description of the methodology used to generate the data in this thesis from the tissue sampling stage to the creation of the paired-short-read sets. Finally, Section 4.5 is a description of the post-processing steps to prepare the data for the experiments, as well as the descriptions of the experiments. Figure 4.10 shows a high level overview of the post-processing steps, and the different experiments and verifications performed with the data. All scripts for SepSIS are present at "`https://github.com/MatthewWaldner/sepsis`" and a list of these scripts is present in Appendix B with a brief description for each script.

4.1 Algorithm Overviews

4.1.1 High-Level Overview

SepSIS identifies strain-specific sequences when given a set of reads suspected or known to contain more than one strain of bacteria. The method to achieve this is based upon the structure of the assembly graph produced by SPAdes. An example of an assembly graph is shown in Figure 4.1. While the assembly graphs produced by SPAdes are a great deal larger and more interwoven than the example, Figure 4.1 shows how alternate forms of full sequences can exist due to differing internal subsequences that start and end at branch nodes.

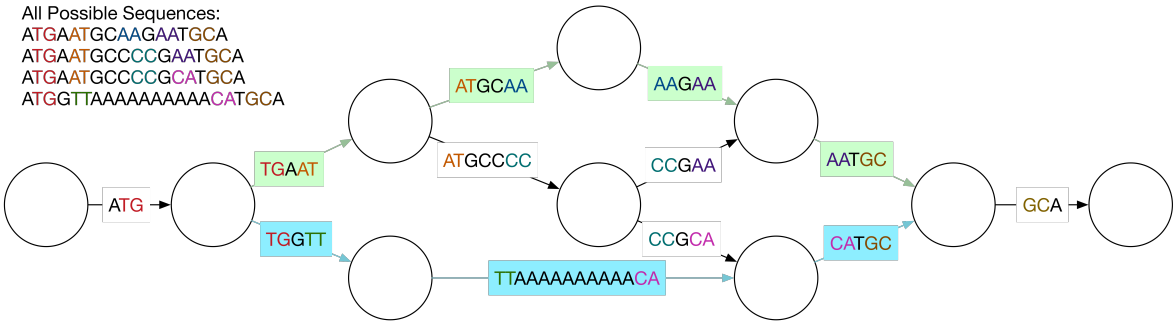

Figure 4.1: Assembly graph, the k-mer overlap is 2, and each unique k-mer overlap is coloured. All possible sequences from simply traversing the assembly graph are shown in the top-left corner. They do not necessarily represent SepSIS output. The blue and green coloured edges are used in an example of the SepSIS output in Section 4.1.1.

The alternate paths possible through the assembly graph correspond to possible candidates for strain-specific sequences.

However, identifying and extracting alternate sequences is not enough to prove they are strain-specific. SepSIS is built to use two methods to classify these sequences as strain-specific or strain-independent. One method is based on relative coverage levels in the assembly graph. This is the experimental method created in accordance with Section 3.1.1. The coverage-based method is further divided into two modes. The difference in the two coverage-based modes is whether strain-specificity is calculated using Z-Scores, or using percentiles. The differences between the coverage-based modes will be discussed further in Section 4.3.6. The third classification method is the verification method and it relies on the provenance of the reads that were merged to create each subsequence in an assembly graph. By checking the isolate of origin of each of the reads comprising a subsequence, it can be determined if reads from only one isolate (and ideally strain) were used to construct a subsequence during SPAdes assembly. This is performed by mapping the reads against the assembly graph produced by SPAdes using minimap2 [22]. The BAM file produced by minimap2 contains the placement of reads relative to each subsequence in the assembly graph. This is necessary because information on identity of reads comprising subsequences in an assembly graph is lost during SPAdes assembly. This is the verification method discussed in Section 3.1.2. For reference, a high-level abstraction of these steps is represented in Figure 4.2.

The objective discussed in Section 3.1.4 is to ensure that the strain-specific sequence has strain-independent subsequences on one or both ends. This is achieved by checking subsequences adjacent to the strain-specific subsequences for the opposite of the strain-specificity criteria. If this criteria is satisfied, the strain-independent subsequences are appended to the end or ends of the strain-specific subsequences.

The output from SepSIS is placed into three files: one file containing the strain-specific sequences with strain-independent subsequences present on both ends of the strain-specific subsequences, one file with

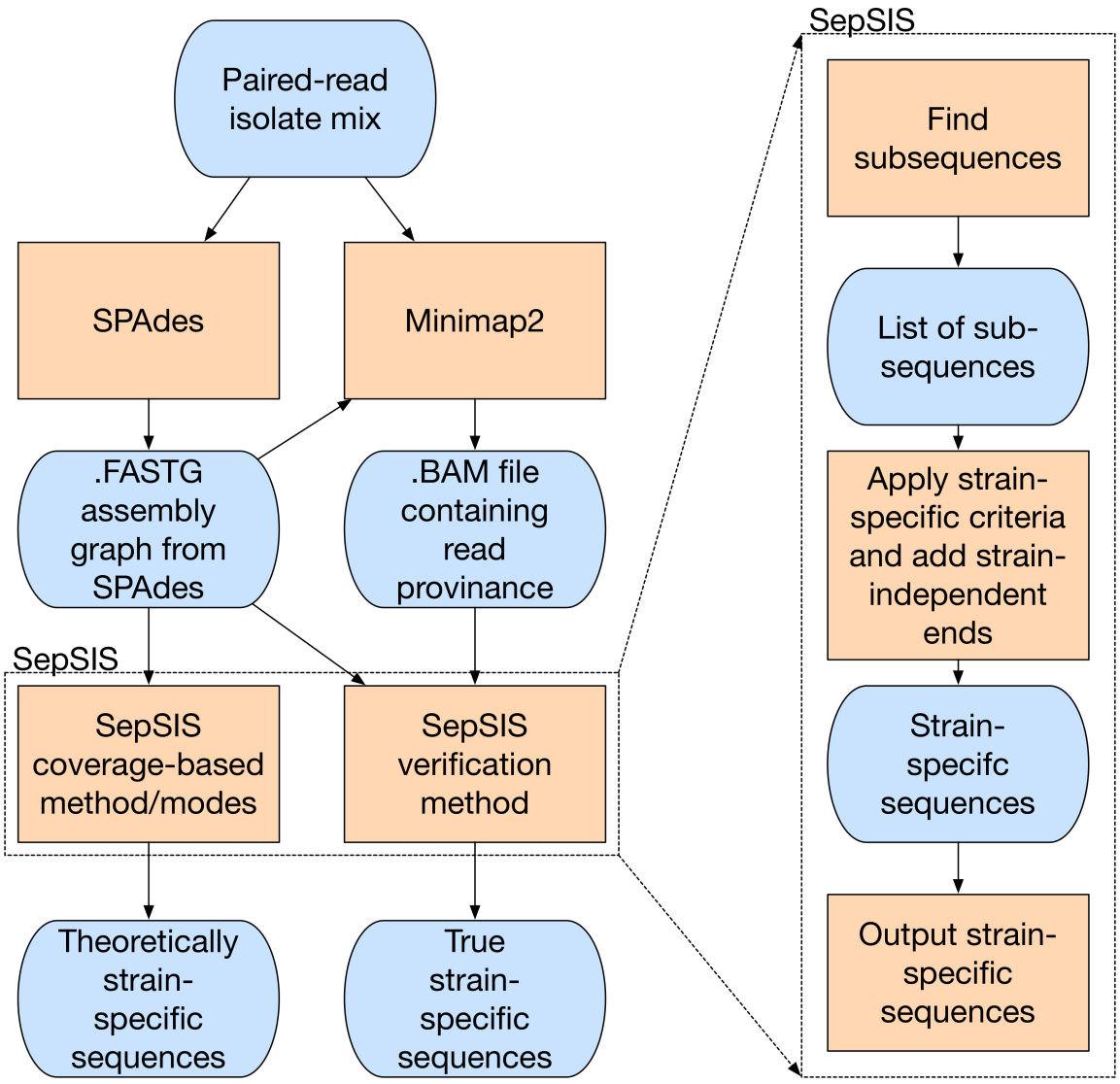

Figure 4.2: A high-level abstraction of the preprocessing and processing steps within the SepSIS pipeline. Pipeline steps are represented on the left while an abstraction of the internal SepSIS steps is represented on the right. The internal steps shown are present in the coverage-based method and the verification method. The differences between the coverage-based modes and the verification method are within the "Apply strain-specific criteria and add strain-independent ends" and "Output strain-specific sequences" steps. Blue objects in the graph represent files and data, while orange objects represent the algorithmic steps.

strain-independent subsequences at the beginning (front end) of the strain-specific subsequences, and one file with strain-independent subsequences on the terminating (back) end of the strain-specific subsequences. Further details of the output format are discussed in Section 4.3.2. An example of possible output sequences follows. Suppose in Figure 4.1 the paths highlighted in blue and green are classified as strain-specific sequences. Then the output from SepSIS would be: `ATGAATGCCCC`, a sequence represented by a path that terminates at the strain-independent `ATGCCCC` subsequence; `ATGAATGCAAGAATGCA`, the top path; and `ATGAATGTTAAAAAAAAAACATGCA`, the bottom path.

4.1.2 Mid-Level Overview

The design of SepSIS relies on taking the proper preprocessing steps. The preprocessing steps are shown in Figure 4.3. Different sets of preprocessing steps are taken based on whether the isolates being examined are combined *in vitro* or *in silico*, and whether one of the coverage-based modes or the verification method is used. The reads for all datasets are trimmed before use in SepSIS using Trimmomatic [6]. This removes low quality nucleotides from the ends of any reads that do not meet a quality threshold. The details of this threshold are further described in Section 2.7.1. After trimming, reads comprising *in silico* mixes are tagged with an identifier representing the isolate of origin at the beginning of each read header. The reads are then mixed by combining the read sets in predetermined combinations. The *in silico* and *in vitro* mixes are then assembled with SPAdes, producing an assembly graph for each mix. The tagged *in silico* mixed read sets are then aligned against the assembly graph using minimap2. These steps provide all the input needed for SepSIS.

Once all the options are set and the preprocessing is finished, this data is passed to SepSIS where multiple internal algorithms act on the data. Two of the input parameters for SepSIS determine which internal algorithms SepSIS runs. The first parameter ("RUNMODE") sets whether a coverage-based mode or the verification mode is used to determine strain specificity. The second parameter ("SUBMODE") sets which of the components of the assembly graph are analyzed. SepSIS can analyze either the whole graph, the CSCCs, or the ISCCs. These components are discussed in background Section 2.7.1. The sections of the assembly graph that are recommended to analyze are the CSCCs, since the CSCCs having a much higher coverage than the ISCCs. This is because the ISCCs in the assembly graph often consist of small and isolated subsequences that are more likely to be misassemblies, while CSCCs occur due to the relatively high coverage and connectivity of the assembled reads. However, all components are available for analysis for the sake of comparison.

Within the "SepSIS" algorithmic step in Figure 4.3., there are 8 distinct steps listed below that serve as an introduction to the SepSIS algorithm. Each of these steps are described in further detail with examples in Section 4.3, beginning at Section 4.3.3.

1. The structure of the assembly graph is altered to make computation simpler, and the components being analyzed are collected into a list. The list of components contains all the components in the assembly

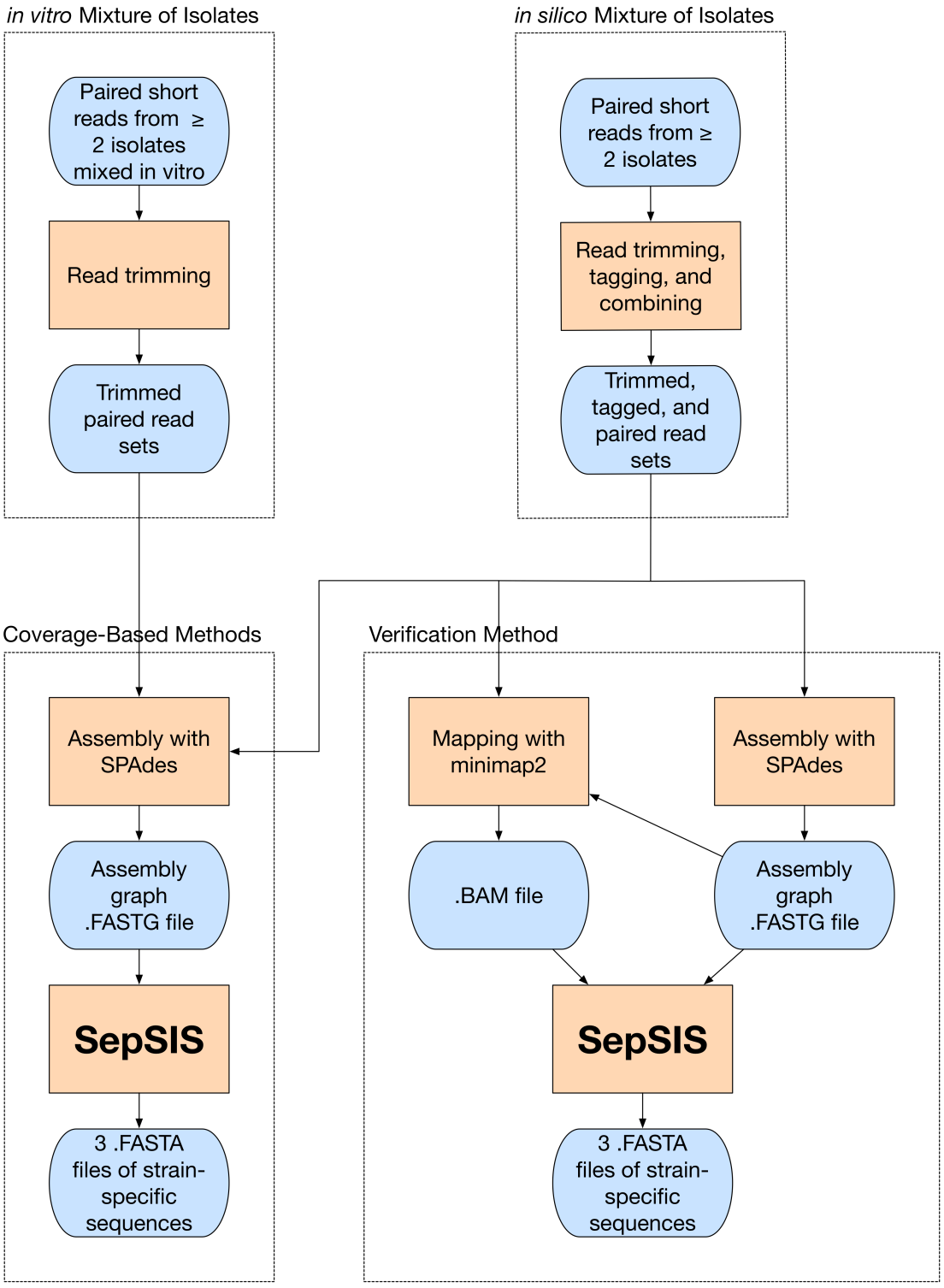

Figure 4.3: The preprocessing steps for SepSIS. The stages are divided by whether the isolates being examined were combined *in vitro* or *in silico*, and whether the coverage-based method or the verification method was used. Blue objects in the graph represent files and data, while orange objects represent the algorithmic steps.

graph, only the CSCCs, or only the ISCCs. This step is based on the option (SUBMODE) selected by the user.

2. The components in the list are searched for branch nodes and terminal nodes, if terminal nodes are applicable. A list of all paths between all branch nodes is created. If the whole graph or the ISCCs are being analyzed, paths between branch nodes and terminal nodes are included as well. The descriptions of these nodes can be found in Section 2.7.1.

3. The list of paths is checked for any two paths with a branch node that is the front of the first path and the back node of the second path. These paths are merged into a single longer path, if the branch node passes the strain-specificity criteria.

4. The paths produced during the previous step are traversed to isolate and extract sub-paths that consist only of nodes that contain strain-independent subsequences on one or both ends, and strain-specific subsequences described by the middle nodes. As described in Section 2.7.2, the term node in an altered assembly graph is synonymous with the subsequence contained in a single incoming edge in a proper assembly graph. Additionally, there is a length threshold that is applied at this stage to limit the total sequence length.

5. The paths with a strain-independent subsequence on only the front of the path are compared against the paths with a strain-independent subsequence on only the back of the sequence. If the strain-specific nodes overlap such that the subsequences could be merged, they are. The pre-merge version of the subsequences that were merged with another are then removed from further processing.

6. The remaining sequences with a strain-independent subsequence on only one end (one-ended sequence) are compared to the sequences with strain-independent subsequence on both ends (two-ended sequence). If a one-ended sequence is a proper ordered subset of a two-ended sequence, the one-ended sequence is removed from its list. This removes any incomplete, partial-duplicate, one-ended sequences.

7. Any fully duplicate sequences are removed from the pending output. The duplicates are removed at this step instead of during sequence extraction or merging in order to reduce the time complexity and run time of the previous steps.

8. Lastly, the lists of nodes are converted to nucleotide sequence strings and output into three files, depending whether the ends have strain-independent subsequences on both ends, on the front (starting) end, or on the back end. Note that 5' - 3' and 3' - 5' sequences are allowed to be in the same file using this criterion for sequence separation.

Several post-processing steps are taken to ready the output produced by SPAdes for analysis. These post-processing steps are discussed in further detail in Section 4.5, and can be summarized into the points immediately below.

- The strain-independent subsequences in the SepSIS output are extracted from their respective sequence and a locally downloaded BLASTN tool from BLAST+ version 2.3.0 is used to compare these sequences against all existing (11 total) completed *M. bovis* genomes [9]. (All further uses of the term BLASTN-ed or BLASTN-ing will refer to this local version of BLAST, unless otherwise specified.) Any subsequence that falls below 94% ANI are removed from the parent sequence, and sequences with no remaining strain-independent subsequences are removed from further analysis. This is performed to ensure that the strain-independent subsequences were common to at least one of the completed *M. bovis* genomes.

- Contamination with *Stenotrophomonas maltophilia* was a problem in previous *M. bovis* read sets not used in this thesis, but grown in the same lab. Any sequences in the output of SepSIS deemed to be contaminated with *S. maltophilia* are removed.

- The validation method was shown to produce duplicated "strain-specific" sequences due to SPAdes assembling the same sequences multiple times. These untrue "strain-specific" sequences were removed from further processing.

As discussed in Section 3.3 and Section 3.4, there are goals for the verification and experimentation. The experiments conducted using the post-processed SepSIS output to satisfy these goals follows:

- The coverage-based output is compared against the validation method output for both the *in silico* and *in vitro* isolate mixes (Section 3.3.1).

- The ability of SepSIS to successfully output strain-specific sequences in larger mixes is evaluated (Section 3.3.2).

- The existence of multiple strains of *M. bovis* on a single culture plate is investigated (Section 3.4.1).

- The effect of purposeful contamination in SepSIS mixes on the SepSIS pipeline is investigated (Section 3.4.2).

- Paired lung and joint *M. bovis* isolates are investigated for subsequences associated with the particular tropisms (Section 3.4.3).

4.2 Data Preprocessing

Different preprocessing steps are necessary depending on whether isolates are mixed *in vitro* or *in silico*. The preprocessing steps are represented in Figure 4.3. For isolates mixed *in vitro*, the preprocessing consists only of read trimming in accordance with standard short-read assembly practice. Trimming is performed with Trimmomatic v0.36 with the following settings: "SLIDINGWINDOW:5:15 LEADING:5 TRAILING:5 MINLEN:25" [6].

For datasets generated by mixing reads *in silico*, the reads are trimmed with Trimmomatic v0.36 using identical settings as above. Each read within a sequenced isolate is marked with an identifier at the beginning of the read header, signifying the isolate of origin for that read. For example, the read:

```
@M04229:68:000000000-BVYJN:1:1101:17792:1964 2:N:0:24
```

has the sample ID "MPLM45" added to it making it:

```
@MPLM45_M04229:68:000000000-BVYJN:1:1101:17792:1964 2:N:0:24
```

The script to accomplish this is named "AddSampleNameToReads.py", as listed in Appendix B. The final step is to concatenate the multiple paired-read files into two files, with one file containing forward (5') reads with R1 in the name to designate the direction and one file containing backward (3') reads with R2 in the file name.

The next step varies based on whether the coverage-based method or the verification method is used. The coverage-based method requires the combined reads from either isolate mixing method to be assembled using the *de novo* assembler SPAdes with additional settings "-k 55 −−careful" [5]. Note that the "-k" option dictates the k-mer length during assembly, and also the overlapping nucleotide length in the output assembly graph. This value of 55 was chosen to provide a small consistent value allowing for varying sizes of input reads to be assessed. This also stops the k-mer length from being a point of variation between runs of SepSIS, allowing for a simpler comparison of output sequences. The assembly graph .FASTG file produced by SPAdes is the the sole input for the coverage-based algorithm in SepSIS.

The verification method requires a SPAdes assembly step identical to the coverage-based method above. In addition to this step, minimap2 maps the *in silico* combined and tagged reads against the assembly graph [22]. The position of each of the tagged reads is represented in the output BAM file relative to the assembly graph, allowing for the determination of the strain-specificity of each subsequence within the BAM file. The output BAM file from minimap2 has to be sorted and indexed by SAMtools to allow for access in SepSIS [23]. This step is available in the file "CreateBAMFilesForContigs.py" (Appendix B). The verification method requires both the assembly graph from the SPAdes and the .BAM file from minimap2 and SAMtools as input.

4.3 SepSIS

4.3.1 Script Structure and Input Variables

The core of SepSIS consists of 3 scripts: "SepSIS.py", "recycle_utils.py", and "utils.py", listed in Appendix B and available at https://github.com/MatthewWaldner/sepsis. The "SepSIS.py" script serves as the main file for user interaction. It takes in input variables, calls functions from other scripts, and outputs

the final strain-specific sequences. The script "`recycle_utils.py`" contains 5 short utility functions taken from a SPAdes add-on utility named Recycler that interact with an assembly graph [41]. These functions would have only had changes in coding style and not purpose had they been written anew for this thesis. The script "`utils.py`" contains all of the custom built functions for the SepSIS algorithm.

`SepSIS.py` requires a total of 7 input parameters when run. The parameter "−−RUNMODE" has 3 possible values for the user to select whether a coverage-based approach or the verification approach is used to evaluate the assembly graph for strain-specific sequences. Two coverage-based approaches (modes) are built into SepSIS. The precise functions are described more in Section 4.4.3, but a short summary of the modes follow. The first mode, named "ORGANIC_Z", utilizes an evaluation of Z-Scores calculated from the coverage of the nodes in the assembly graph to determine if a node is strain-specific. The second coverage-based mode, "ORGANIC_P", instead uses a percentile-based evaluation of assembly graph node coverage. The verification method is called "SYNTH", named after the use of synthetically mixed reads for verification.

The parameter "−−SUBMODE" dictates which parts of the assembly graph are analyzed by SepSIS. The options to "−−SUBMODE" are: "CYCLIC", which performs strain-specific sequence isolation on only the CSCCs of the assembly graph; "ISOLATED", which performs strain-specific sequence isolation on the ISCCs; and "BOTH", which performs strain-specific sequence isolation on both components and includes the corner cases between the two. These components are discussed in Section 2.7.1.

Several parameter settings specific for sub-algorithms within SepSIS. The parameters "−−Max_Score_Value" and "−−Min_Score_Value" are variables that act as threshold values for each "−−RUNMODE" to determine strain-specificity, and will be discussed in-depth in Section 4.3.6. The parameter "−−kmerLength" sets the length of the k-mer overlap found in the assembly graph. This parameter's input must match the single "-k" parameter input used by SPAdes to allow the assembly graph to be interpreted properly. For this thesis "-k" in SPAdes and "−−kmerLength" are set to 55 because a mid-length k-mer value is recommended for use in SPAdes.

Lastly are the parameters that dictate input and output. The input parameter "−−fastgFileIn" must be provided for the .FASTG assembly graph produced by SPAdes. The parameter "−−bamFileIn" requires the BAM file that is needed when the algorithm is run using the verification approach (specified by "−−RUNMODE SYNTH"). The parameter "−−outDirectory" is followed by the path to the output directory for all output files. The parameter "−−outSuffix" is followed by a string of text the user can add to the file name of the output files.

4.3.2 Output Format

For each run, SepSIS outputs 3 .FASTA files (shown in Figure 4.3). SepSIS requires each strain-specific sequence that it extracts to have at least one strain-independent subsequence attached on the front or back end. Therefore, SepSIS outputs one .FASTA file for each possible condition: sequences with a strain-independent subsequence only at the front, designated with "FrontEnds" in the output file name; sequences

with a strain-independent subsequence only at the back, designated with "BackEnds"; and sequences with strain-independent subsequences at both ends, designated with "BothEnds". For clarification, 5' - 3' and 3' - 5' sequences may be present in the same file. Front and back refer only to the literal order of the sequence of nucleotides, not their orientation.

The output files contain .FASTA sequences with headers based upon the headers of the FASTG sequences. Each node in a SPAdes FASTG assembly graph contains at least one connection of incoming and outgoing edges. This is reflected in headers, shown in the following example:

```
>EDGE_18662_length_203_cov_9.378049':EDGE_1040_length_161_cov_14.950000';
```

In this example, the first node represents a 3' - 5' (indicated by the ' after the coverage) subsequence from an incoming edge of length 203 and coverage of 9.378049, to an outgoing edge named `EDGE_1040`. SPAdes assembly graphs are described in more detail in Section 2.9.3.

SepSIS produces output sequences with a unique header format based on a concatenation of the headers from a .FASTG file. An example follows:

```
>EDGE_300722_length_167_cov_19.891304_._EDGE_300664_length_201_cov_31.937500
_._EDGE_299882_length_163_cov_133.809524
```

In this format, the characters "underscore period underscore" (_._) replace the ":" as a separator of the edge information. This is because some programs that use the .FASTA format designate the ":" as a special delimiting character, causing errors. The .FASTA format differs from .FASTG in that it can not contain multiple outgoing edges, but does retain the header information from the .FASTG format. The full length of the sequence following this header will depend on the the k-mer length input into SPAdes and SepSIS. If the k-mer length is set to 55, the sequence following the header above would have length of 421. This length is derived from the total shown length of the nodes (167 + 201 + 163 = 531), minus the two 55 length overlaps (531 - 55 - 55 = 421).

In addition, if the validation method is selected, the header will contain the name of the isolate as added by the user to the reads during the preprocessing stage the name is followed by a triple underscore (___). An example of this is shown below. Note that the line wrap in the example is due to the header being too long to express in this document, and does not appear in the .FASTA file.

```
>MPLM9___EDGE_290302_length_4247_cov_59.894571_._EDGE_21842_length_122_cov_72.000000
_._EDGE_50404_length_140_cov_176.578947_._EDGE_61376_length_174_cov_105.981132
```

4.3.3 SepSIS Algorithm

In Section 4.1.2, a list of 8 distinct steps to SepSIS was given. These steps are further explained in 6 subsections below, with each subsection listed here. In addition, the 3rd and 4th steps utilize the strain-specific criteria functions which are described in Section 4.3.6. Due to a lack of algorithmic complexity in steps 5, 6, and 7 from Section 4.1.2, the steps have been compressed into step 5 (The Merging of One-Ended Paths, and the Removal of Proper Subsets of Two-Ended Paths and Duplicate Sequences). The flow of these steps/subsections is presented in Figure 4.4. The orange objects in Figure 4.4 are the same as the steps presented in the list below.

1. Component Separation from the Assembly Graph
2. Identification of All Possible Paths Between Branch and Terminal Nodes
3. Merging of Paths Based Upon Strain-Specific Criteria
4. Splitting of Merged Paths Using Strain-Specific Criteria
5. Merging of One-Ended Paths, and Removal of Proper Subsets of Two-Ended Paths and Duplicate Sequences
6. Sequence Output Conversion

4.3.4 Component Separation from the Assembly Graph

In the first stage, the structure of the assembly graph is slightly altered as briefly discussed in Section 2.7.2. Figure 4.5 shows the product of this alteration. All of the incoming edges are converted to nodes, and connected with edges in a structure functionally matching that of an assembly graph. Subsets of the altered assembly graph are copied into three lists of components: the List of CSCCs, the List of ISCCs and the List of All Components in the Graph. The components are discussed in Section 2.7.1. The python networkx package is used to construct these three lists. The List of All Components in the Graph is generated by iterating through the graph and copying all components to the list. The List of CSCCs is generated by iterating through the graph and isolating all SCCs of size 2 or greater. The remaining SCCs in the graph have a size 1 are identified and placed into the List of ISCCs. Separate graphs consisting of only the CSCCs and only the ISCCs are then created.

4.3.5 Identification of All Possible Paths Between Branch and Terminal Nodes

The next stage is to iterate through the relevant list of components (List of CSCCs, List of ISCCs, or List of All Components in the Graph) to identify the set of branch nodes (in CSCCs) or the set of branch nodes and terminal nodes (in ISCCs or the whole assembly graph). Collectively, the branch nodes and the terminal nodes will be referred to as primary nodes. If the algorithm is set to analyze the ISCCs or the whole assembly

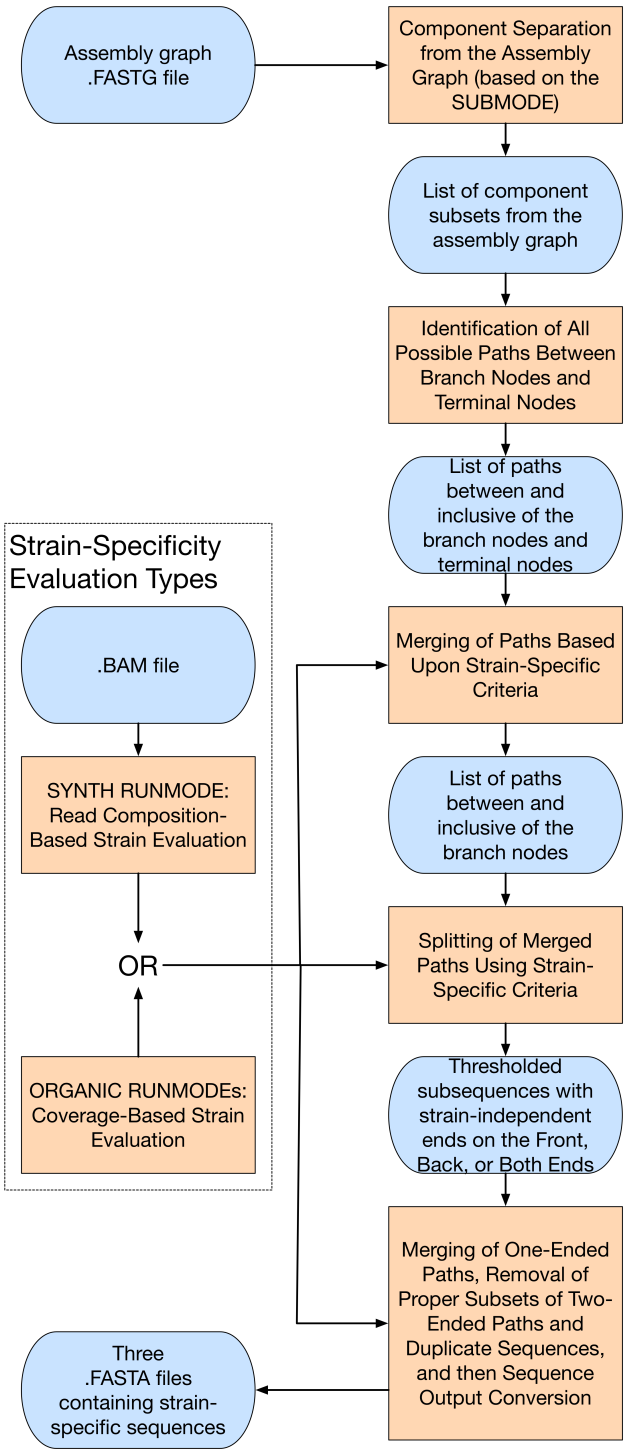

Figure 4.4: The primary internal steps of SepSIS. Blue objects in the graph represent files and data, while orange objects represent the algorithmic steps. The majority of the steps shown in this figure are all contained within the "SepSIS" step in Figure 4.3. The input and output files for SepSIS are also shown. Note that steps listed in Section 4.3.3 and the titles of Sections 4.3.4 to 4.3.10 match the orange steps on the right of the diagram and the "Strain-Specific Criteria" box on the left side of the diagram.

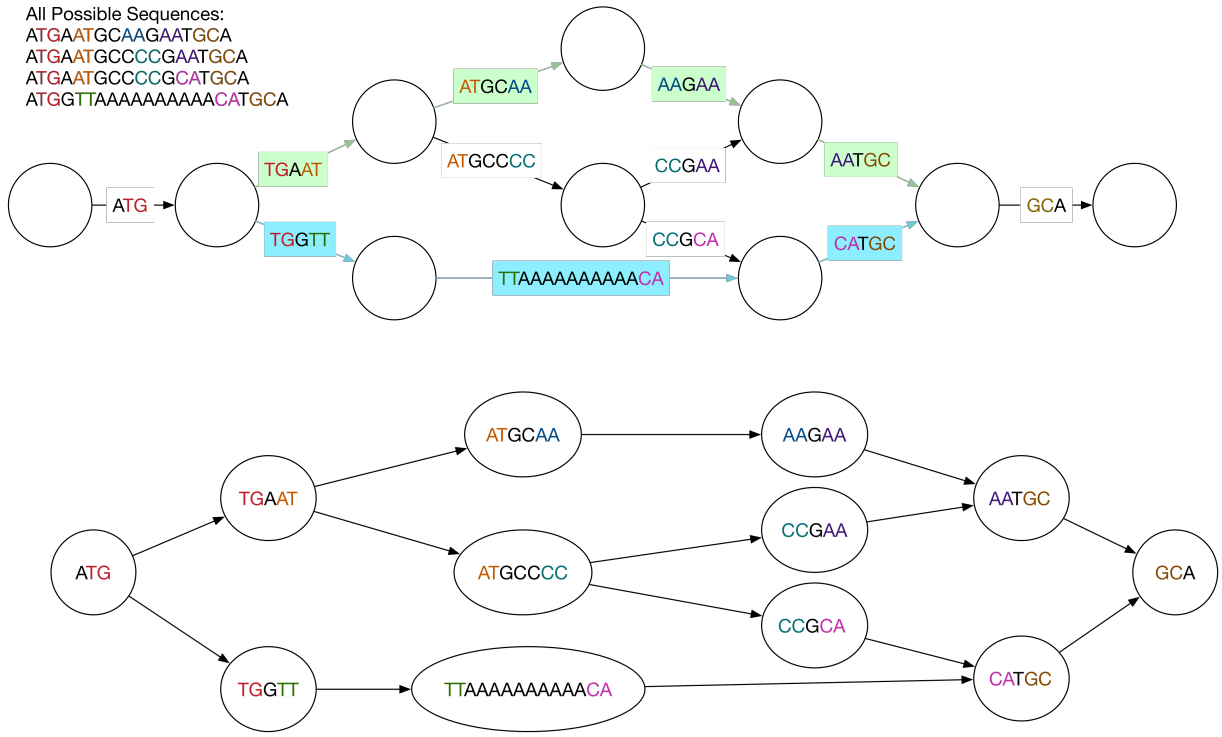

Figure 4.5: An altered assembly graph contrasted with the assembly graph used to create it. The altered assembly graph is the bottom graph and it contains the same information as in the assembly graph in Figure 4.1, which is included at the top of the figure. The k-mer overlap is 2, and each unique k-mer overlap is coloured.

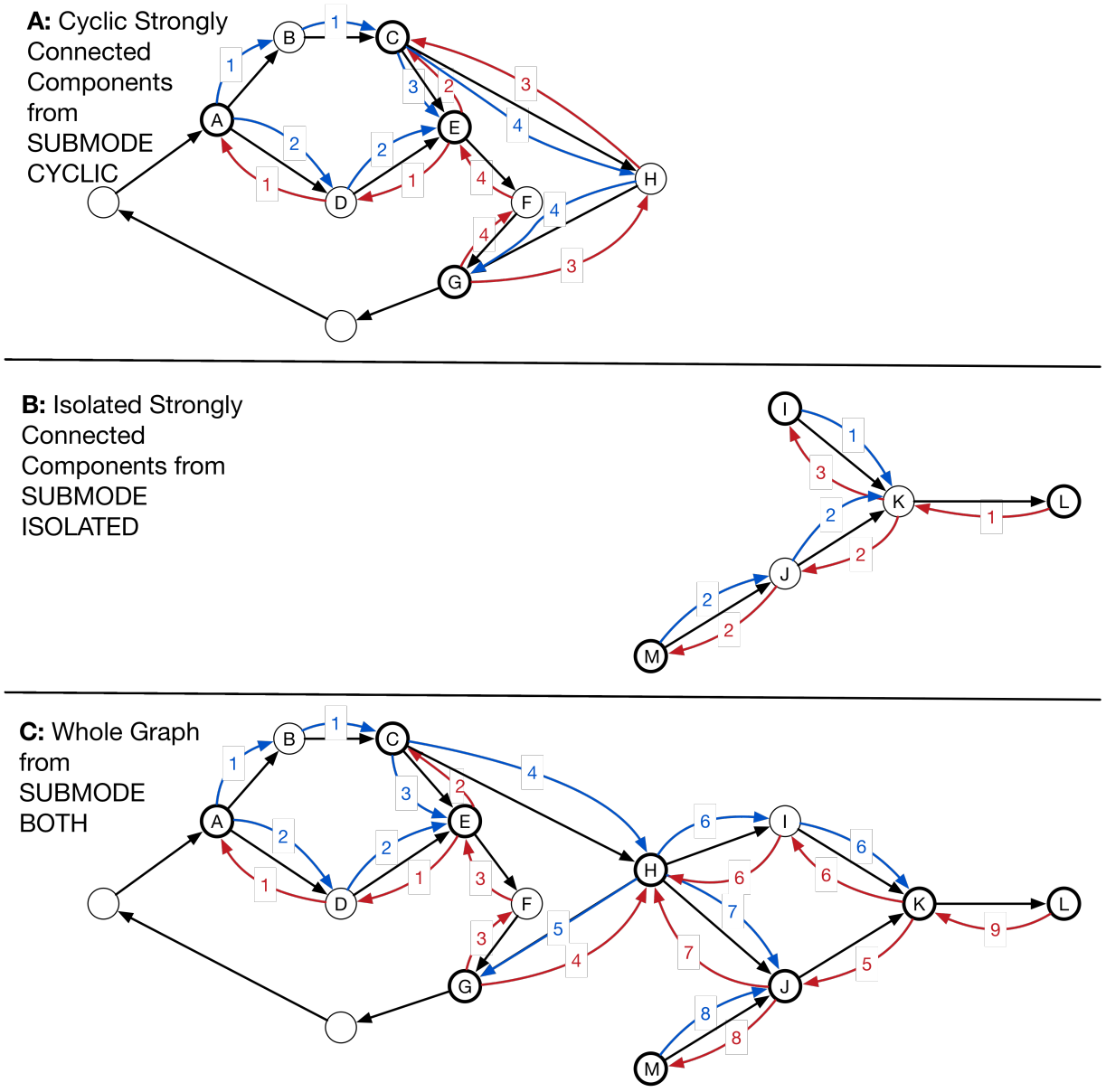

Figure 4.6: The identification of all possible paths between primary nodes in each of the 3 SUBMODEs. Primary nodes are bolded. Forward traversals are marked in blue, while reverse traversals are marked in red. The steps of each individual traversal share an identifying number. The paths created for each graph subset/SUBMODE are as follows:
CSCC/CYCLIC Traversals (panel A): [ABC, ADE, CE, CHG, EFG]
ISCC/ISOLATED Traversals (panel B): [KL, MJK, IK]
Whole Graph/BOTH Traversals (panel C): [ABC, ADE, CE, CH, HG, HIK, HJ, JM, EFG, KL, KJ]
Duplicate paths created by the forward and reverse traversals are removed. Note the inclusion of novel paths surrounding the node H in the whole graph. Also note the blank nodes. In theory, the subsequences in these nodes would be shared by all possible assemblies as there are no alternate paths around them. Therefore, they are not assessed for strain-specificity.

graph, branch nodes and terminal nodes are identified. If the algorithm is set to analyze the CSCCs, only the branch nodes are found. Identification of these nodes is performed by checking the number of successor and predecessor nodes for each node in a component. The nodes with multiple successors, multiple predecessors, no successors or no predecessors are all added then to a list named the List of Primary Nodes. For reference, these will be referred to as successor nodes, predecessor nodes, end nodes, and start nodes, respectively.

After the List of Primary Nodes is accumulated, the next step is to create paths between all primary nodes in the subgraph being analyzed. Visualizations of these steps are presented in Figure 4.6 for iterations through a CSCC in panel A, the ISCCs in panel B and the whole graph in panel C. For each successor node in the List of Primary Nodes, the subgraph is recursively traversed forward from each of its successors, until the traversal encounters another primary node. The recursive function then returns all possible paths forward that terminate in another branch node. In Figure 4.6, a single iteration of this would be represented by blue traversals 1 and 2, creating paths ABC and ADE. These paths are all added to a list named the List of All Possible Paths (LAPP).

The next step is to traverse the subgraph backwards from each predecessor node in the List of Primary Nodes, until the traversal encounters another primary node. The recursive function then returns all possible paths between the relevant primary nodes. The returned path is in the forward direction, despite the reverse traversal. From node E in Figure 4.6, the returned paths would be ADE and CE. If the algorithm is analyzing the ISCCs or the whole graph, identical forward and reverse traversals occur with the start and end nodes, respectively. An additional note to make is that during the traversals, the subsequences in each node must be in the same direction, for example 5' to 3', to be added to the path. This ensures consistent sequence directionality in the paths. All returned paths are added to the LAPP, and then all duplicate paths are removed from the LAPP. The duplicate paths are removed at this stage because this results in less time complexity than monitoring the new insertions to the LAPP for duplicates as it is being constructed.

4.3.6 Strain-Specific Criteria

There are 3 methods of assessing whether or not each subsequence is strain-specific. Each of these methods has a specific set of criteria for determining whether a subsequences is strain-specific. The method used depends on the chosen RUNMODE. These sets of criteria are based on either relative coverage values (the coverage-based method/modes) or on the percent composition of reads from the isolates within the mix (the validation method). In theory and assuming level coverage across a set of strain-independent assembled subsequences, the strain-specific subsequences within a mix of isolates will have a coverage of approximately (Strain-Independent Subsequence Coverage / Number of Isolates in Mix).

Based on this concept, the ORGANIC_Z mode calculates the mean and standard deviation of the coverages for the subgraph being assessed. These values are then used to calculate a Z-Score whenever a node (representing a subsequence) is assessed for strain-specificity. The Max_Score_Value and Min_Score_Value input parameters are used at this point as well. These values act as upper and lower Z-Score thresholds by

which a node is designated strain-specific. If the calculated Z-Score is less than or equal to the maximum Z-Score cutoff and greater than or equal to the minimum Z-Score cutoff, the subsequence represented by the node is designated strain-specific. The upper and lower thresholds act as boundaries to find the strain-specific sequences, while ideally blocking out misassembled subsequences and strain-independent subsequences.

However, Z-Scores are meant to work with normally distributed data. Unfortunately, during development of SepSIS it was determined that the coverage values for assembly graph subsequences are not consistently normally distributed. This was definitively determined by examining where the reads of a set of testing isolates mapped to a reference genome. The reads mapped into peaks and troughs throughout the genome, without consistent patterns between genomes. Therefore, the ORGANIC_P mode was implemented. ORGANIC_P is almost identical to ORGANIC_Z, except it calculates a percentile for each subsequence coverage value based on the median coverage value for all nodes. Max_Score_Value and Min_Score_Value are interpreted as the maximum and minimum percentile thresholds within which a isolate is considered strain-specific. This allows for assessment of the non-normally distributed coverage values. The use of these maximum and minimum coverage thresholds, which are determined based on user input and on the particular subgraph being assessed, are design decisions made to compensate for the highly variable coverage in *M. bovis* data, as discussed in Section 3.1.3.

The SYNTH RUNMODE is the verification algorithm discussed in Section 3.1.2 and is implemented differently from the coverage-based modes. When the function that verifies strain-specificity is passed a node, it isolates the node header from the node data. The data from the input BAM file is then searched using the node header and the specific reads mapping to that node are fetched. By using the sample ID prefix at the beginning of each read, the sample with the majority of reads mapping to the node subsequence is identified, as well as the ratio of reads belonging to the majority sample ID. The Max_Score_Value and Min_Score_Value parameters act as the maximum and minimum thresholds for the subsequence to be considered strain-unique. All experiments in this thesis use a threshold of 1, meaning that all reads mapping to the node must belong to a single sample ID for the node to be considered strain-specific.

4.3.7 Merging of Paths Based Upon Strain-Specific Criteria

The merging function is given the List of All Possible Paths (LAPP) as described in Section 4.3.5. The LAPP contains all paths inclusively between all branch nodes or inclusively between all primary nodes, depending on the section of the graph being analyzed. The merge function will perform the following actions described in this subsection. It will initially record the first entry in the LAPP, which will be referred to as PATH1. PATH1 is initially checked for any repeating nodes. If PATH1 has repeating nodes, it is removed from the LAPP. Note that DNA repeats of greater than a few hundred nucleotides (the size of a small subsequence represent by a node) do occur naturally in genomes. However, node repeats cause small sequence loops, and therefore are excluded from analysis. PATH1 is also checked for a length longer than one node. If the PATH1 has a length of only one node, PATH1 is removed from the LAPP. If PATH1 is removed from the LAPP, the

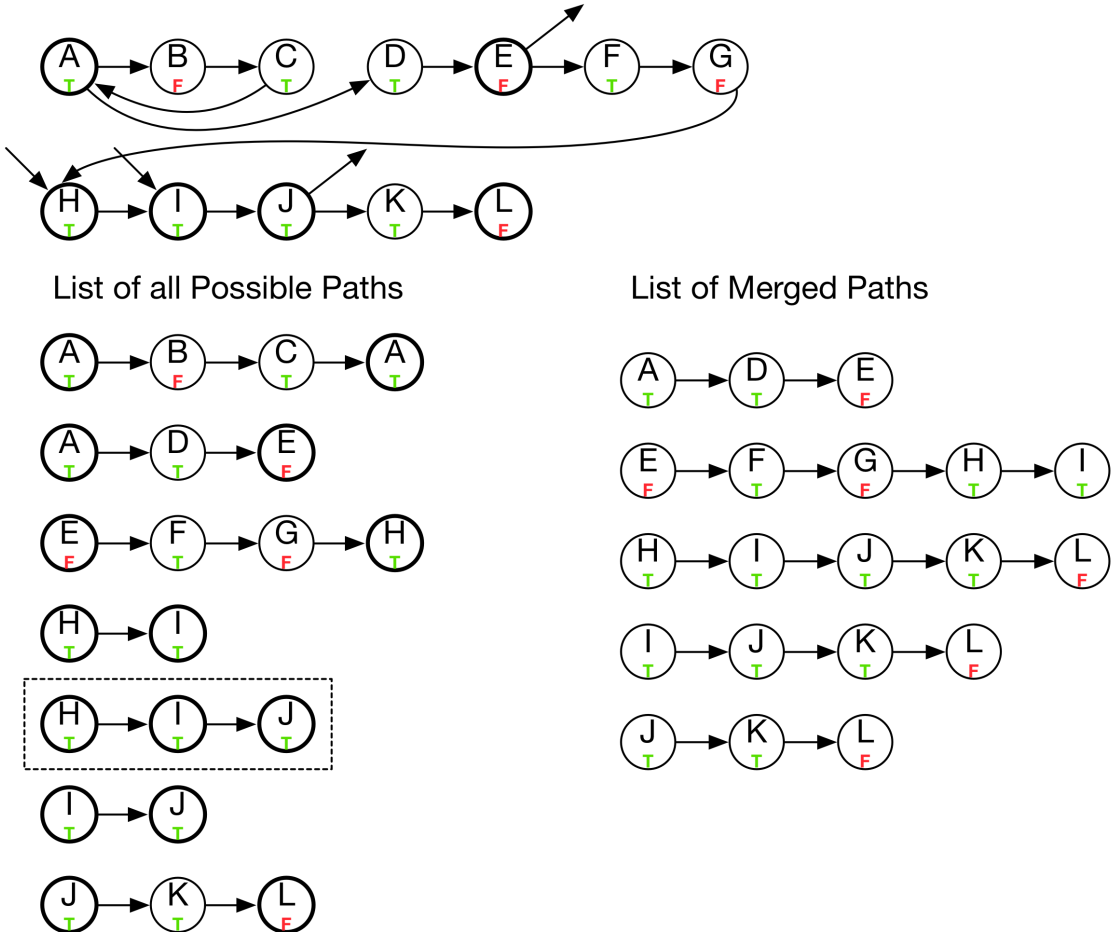

Figure 4.7: An input List of All Possible Paths (LAPP), and an output List Of Merged Paths (LOMP) for the Merging of Paths Based Upon Strain-Specific Criteria step. An example of the graph containing the listed nodes is at the top of the figure. The coloured Ts and Fs in the nodes represent whether the nodes pass or fail the strain-specific criteria, and the primary nodes have bolded circles. Note that some nodes have incoming or outgoing edges with no nodes attached. This is done to create primary nodes for the example, and would not happen in a true assembly graph. In the List Of Merged Paths, the nodes (circles) are not bolded because the distinction between primary and other nodes is not relevant to the algorithm after this point. For this example, the maxPathNodeLength is set to 3. Notable occurrences: The path ABCA is removed due to the presence of a repeating node. ADE is added to the List of Merged Paths due to never encountering another list to merge with. The path HIJ is inserted into the List of All Possible Paths after HI and IJ are merged because it is less than or equal to 3 nodes long.

next path in the LAPP is set to PATH1 and evaluated with the above criteria.

If PATH1 passes the previous criteria, the start and end nodes of the path are evaluated using the designated strain-specific criteria, as discussed in Section 4.3.6. If at least one end node passes, the LAPP is checked for all other paths that share the start and/or end node(s) that passed the strain-specific criteria. If two paths share a valid end node, the second path, named PATH2, is evaluated using the same duplication and length criteria as PATH1. If PATH2 passes, PATH1 and PATH2 are merged. If PATH2 does not pass, it is removed from the LAPP. An exception to this case occurs if the start node of PATH1 matches the end node of PATH2 and the start node of PATH2 matches the end node of PATH1. This would also create a sequence loop, and therefore the paths are not merged if this is the case.

The length of this merged path is then evaluated. If the number of nodes in the path is greater than an internal parameter named maxPathNodeLength, the merged path is added to the List of Merged Paths, referred to as the LOMP. If the merged path is less than or equal to maxPathNodeLength, it is added to the front of the LAPP. For this thesis, maxPathNodeLength was set to 8. For reference, the approximate average length of subsequence represented by a node in an *M. bovis* assembly graph is 1000 bp, and the average length of a gene in the *M. bovis* PG45 reference genome is 1058 bp [50]. This cutoff allows for sequence strings to be generated that are able to contain, based on the average, approximately 8 adjacent genes. This is meant to prevent a single sequence in a potential output sequence from being truncated, but also prevents long runtime loops in the case of highly looped or branched structure in the assembly graph. Note that paths of longer than 8 nodes can be added to the LOMP though the combination of longer pre-existing paths in LAPP. A nucleotide length based cutoff was not chosen due to the high variation in the sequence length contained with nodes. The remaining paths in LAPP are checked for matching end nodes. If PATH1 never encounters another list that it can merge with, it is also added to the LOMP. An example of the LAPP and LOMP are represented in Figure 4.7.

4.3.8 Splitting of Merged Paths Using Strain-Specific Criteria

In this stage, the LOMP is passed to 3 separate functions that evaluate each path in the LOMP and split it based on specific criteria, as seen in Figure 4.8. The first function is the BothEnds splitting function. This function iterates through each path in the LOMP and extracts subpaths that have strain-independent subsequences on both ends of a strain-specific sequence. These subpaths are placed in the List of Both End Paths. The second function is the FrontEnds splitting function, and it extracts all subpaths from paths in the LOMP that have a strain-independent subsequence on the front end of the path, but not on the back end, and adds the subpaths to the List of Front End Paths. This can be seen in Figure 4.8 with path GHI. The third function is the BackEnds splitting function. It extracts all subpaths from the path in the LOMP that exclusively end in a strain-dependent subsequence and adds them to the List of Back End Paths. All paths that are not bordered by a strain-independent subsequence are excluded from further steps.

These functions are all similar in purpose but differ enough to mandate separate implementations. All

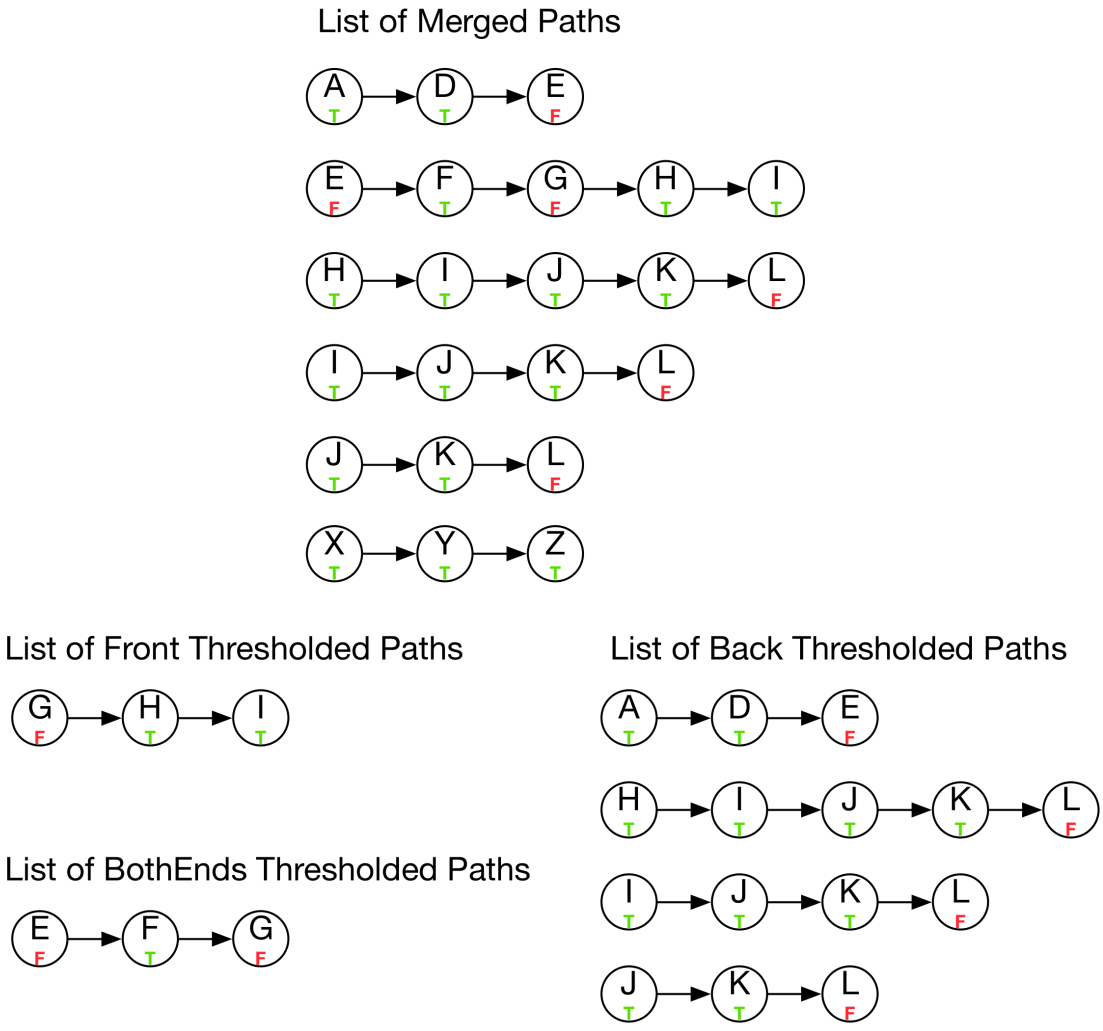

Figure 4.8: The input List of Merged Paths (LOMP) and output Thresholded Paths Lists for the Splitting Merged Paths Using Strain-Specific Criteria step. This example is continued from Figure 4.7, with the addition of the path XYZ. The path XYZ has been added to the List of Merged Paths to show that it would be removed from future analysis due to not having a strain-independent start or end node. The coloured Ts and Fs in the nodes represent whether the nodes pass or fail the strain-specific criteria. Notable occurrences: The path GHI is present in the FrontEnd Thresholded Paths list, despite node G starting in the middle of the path EFGHI.

three functions use a 2-node forward sliding window to check the strain-specific pass or fail conditions of two adjacent nodes in the path being analyzed. For each function there are criteria for sections of the path to continue to be considered for their respective type lists. The BackEnds function is the simplest and makes the best example. A path being assessed by the BackEnds function can never start with a strain-independent node, so if the sliding window ever encounters a strain-independent node, the traversal of that path will end with that node. It will then return a subpath if it has found a valid one. Another example is that a BothEnds path can not be of size 2 while the other two types can. These functions become more complex when the SYNTH RUNMODE is chosen. The SYNTH functions must also take into consideration the identity of the majority read for each node to ensure that two nodes comprised entirely of reads from two different isolates are not assigned as specific to the same isolate and output in a single path. The strain-independent node must be check to ensure reads from more than 1 isolate comprise it, rather than simply failing the strain-specific criteria.

4.3.9 Merging of One-Ended Paths, and Removal of Proper Subsets of Two-Ended Paths and Duplicate Sequences

The final steps are list traversals checking for various conditions. The List of Front End Paths is iterated through, with all but the first node of each path being checked against the front nodes of each path in the List of Back End Paths. If there is a match such that the paths could be combined, the paths are merged and added to the List of Both Ends Paths. The Back End Path that was merged with is copied to a Removal List for later removal from the List of Back End Paths. Once a Front End Path has been checked against all Back End paths, it is removed from the list if it was merged with any Back End path. If it did not match, it remains on the List of Front End Paths.

The next step is to remove any Front End or Back End paths that are subsets of a larger path in the output lists. This is performed by first iterating through the List of Front End Paths and comparing the strings of node names to all entries within the List of Front End Paths and the List of Both Ends paths. This process is then performed again, substituting the List of Front End Paths for the List of Back End Paths. If the iterator path is a proper subset of another path, the proper subset path is removed from the relevant list. Finally, the paths within each list are compared to remove any duplicate sequences. These steps are represented in Figure 4.9.

4.3.10 Sequence Output Conversion

Finally, the List of Front End Paths, List of Back End Paths, and List of Both End Paths are sent to a function to convert each path of nodes into a sequence in the output format discussed in Section 4.3.2. The sequence headers are concatenated in order to form the .FASTA output header. If the validation method (SYNTH RUNMODE) is being used, the isolate identifier is concatenated to the front of the header. The output .FASTA sequence is formed by iteratively concatenating all the .FASTG node subsequences while

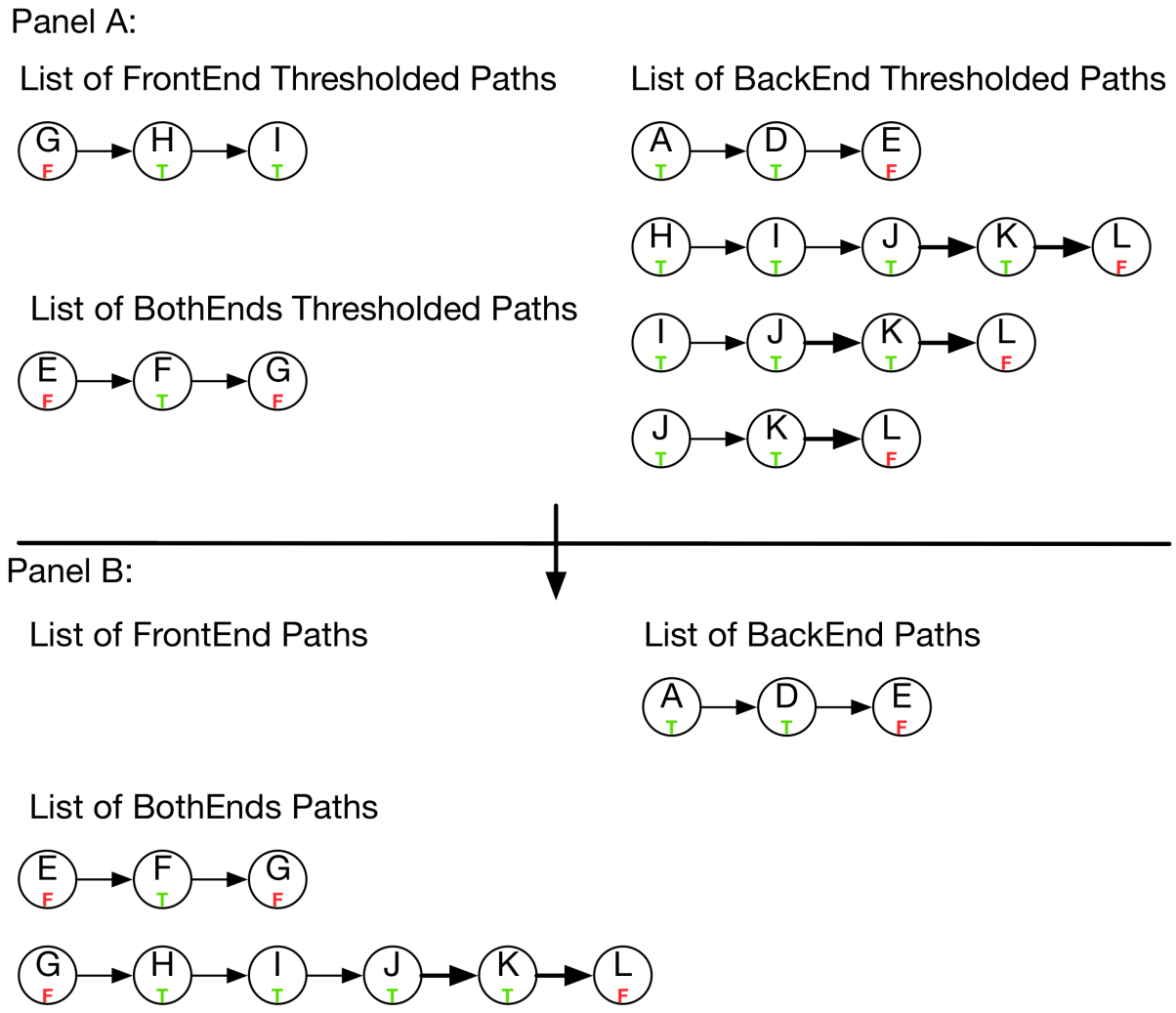

Figure 4.9: The input and output of the Remerge Paths and Remove Subsets steps. This figure continues the examples from Figure 4.8 The coloured Ts and Fs in the nodes represent whether the nodes pass or fail the strain-specific criterion. In the transition from Panel A to Panel B, the FrontEnd Path GHI is checked against all BackEnd Paths. GHI is merged with the paths HIJKL and IJKL, adding two copies of the path GHIJKL to the List of BothEnds Paths. This results in the removal of the paths GHI, HIJKL, and IJKL from their respective lists. There are no FrontEnd or BackEnd paths that are subsets of a of another path, and so none are removed. Lastly the duplicate GHIJKL path is removed, creating the results seen in Panel B.

removing the k-mer length of nucleotides from the front of each subsequence except the first. This is due to the k overlapping nucleotides between the subsequences represented by the nodes. The .FASTA sequences are then written to the three output files.

4.4 Dataset Development and Description

4.4.1 Sample Collection and Growth

During the summer of 2017, the author assisted in the sampling, culture growth and isolation, and DNA extraction of approximately half of the *Mycoplasma* samples used in this thesis under the supervision of Karen Gesy, as discussed in Section 3.2.1. Karen Gesy handled this process for the other half of the samples. It was necessary for some of these samples to be regrown and re-isolated due to poor DNA quality or due to contamination, primarily with *Stenotrophomonas maltophilia*. Additionally, Karen regrew 6 isolates of *M. bovis* stored in an -80°C freezer for sequencing. These isolates were previously sequenced and were selected as the 6 isolates for *in vitro* combination due to their availability. Note that *Mycoplasma* species other than *M. bovis* were processed in the manner described above during this time as well.

The majority of the *Mycoplasma* samples used in this experiment came directly from infected tissue. Infected tissue was refrigerated at -20°C until sampling. Sampling was performed in one of two ways. Infected tissue was lanced with a scalpel and swabbed, or the infected tissue was externally seared using a metal spatula heated under a bunsen burner, and was then cut into and swabbed. The external searing reduced the chance of contamination by killing external contaminants that could have contacted the swab. The swab was used to agitate 3 ml of PPLO broth with Penicillin G and Thallium (I) Acetate and left to sit in the broth for several minutes until removal. The broth was incubated for 48-72 hours at 37°C and 5% CO_2. Each of the *Mycoplasma* samples that were previously isolated and frozen at -80°C were thawed. A sterile 10 µl inoculation loop was dipped in the thawed vial. The inoculated loop was then placed into 3 ml of pleuropneumonialike organism (PPLO) broth with Pen G and Thallium and swirled to ensure the bacteria entered the broth. The broth was incubated for 48-72 hours at 37°C and 5% CO_2.

Dilution and plating occurred after the initial incubation. A 8 by 9 deep-well plate was used for the dilution step. Each well was filled with 90 µl of PPLO broth with no Pen G and no Thallium. For each sample, 10 µl of incubated broth was pipetted into a well in the first column and mixed to create a 10^{-1} dilution. Then 10 µl each of the 10^{-1} dilutions was pipetted into the column and mixed creating a 10^{-2} dilution. This process continued for each sample up to a 10^{-8} dilution. For each dilution, 5 µl of the dilution was added to a colony plate divided into six sections and made of PPLO agar with Pen G and Thallium. These plates were incubated for 4 to 14 days at 37°C and 5 % CO_2, with growth checks occurring at least every 3 days. Colony picking was performed by extracting a single isolated colony from a plate using a Pasteur pipette. The colony was placed into 3 ml of PPLO broth with no Pen G or Thallium. The PPLO was incubated for 48-72 hours at 37°C and 5% CO_2.

DNA preparation and extraction was performed using the Qiamp DNA Mini Kit. The 3 ml of PPLO was vortexed, and 1ml of broth was extracted into a 1.5 ml centrifuge tube. It was at this step that the *in vitro* mixes were created, by combining equal amounts of *M. bovis* as measured by the optical density of the extraction. All further steps were identical among the mixed and un-mixed isolates. The tube was centrifuged for 5 min at 300 x g. The supernatant was carefully siphoned off using a pipette and discarded to preserve the pellet. The pellet was resuspended in 200 µl PBS. Both 20 µl of QIAGEN proteinase K and 200 µl of Buffer AL were added to the mix. The mix was pulse-vortexed for 15 seconds and incubated at 56°C for 10 min. Afterwards, 200 µl ethanol was added and immediately mixed with a pipettor. The mix was pulse-vortexed for 15 seconds. The sample was pipetted to a spin column, and centrifuged at 6000 x g for 1 minute. The spin column was placed in a clean 2 ml collection tube and 500 µl Buffer AW1 was added to it. The column was centrifuged at 6000 x g for 1 minute again. The spin column was again placed into a new 2ml collection tube, and 500 µl of Buffer AW2 was added before centrifuging the mix at 18000 x g for 3 minutes. The spin column was placed in a new 2 ml collection tube and spun at full centrifuge speed for 1 minute to remove any remaining solution. The spin column was placed into a 1.5 ml micro centrifuge tube and 50 µl Buffer AE was added. The tube was incubated for 1 minute and centrifuged at 6000 x g for 1 minute. Each of the tubes were sent for sequencing.

4.4.2 Read Set Description

A total of 117 paired short-read sets of *Mycoplasma* species comprised the dataset for this thesis. All of these read sets were produced using the Illumina Miseq sequencing platform, but the sequencing took place at different locations and used different read lengths. Of these short-read sets, 92 were sequenced locally at the Hill Lab in the Department of Veterinary Microbiology in the Western College of Veterinary Medicine at a length of 250 bp. These isolates were given the arbitrary identifier tag MPLM. Dr. Karen Register at the Nation Animal Disease Center, a part of the United States Department of Agriculture, sequenced 20 isolates of *M. bovis* sent to her by our lab, and these isolates were also tagged with MPLM. These isolates were sequenced at a length of 150 bp. The set of previously sequenced six *M. bovis* samples used as the *in silico* baseline for comparison against the *in vitro* mixed isolates had differing sequencing details. Of the isolates, 4 were tagged with MYCO and sequenced at GeneSeek, a Neogen company in Lincoln, NE, USA at a length of 150 bp. The remaining two isolates were tagged with MP00 and sequenced by the National Research Council at the Plant Biotechnology Institute in Saskatoon, SK at a length of 300 bp. Metadata for all of these isolates is available in Table A.1, including the read set group that each isolate is within. Statistics for the independent assembly of each of the read sets are available in Table A.2. The read set groups are further described as follows:

Register Sequences at 150 bp:

A. 20 of the paired-read sets were *M. bovis* isolates sampled from 4 bovine tissue samples and sequenced. These 4 tissue samples consisted of 2 pairs of bovine lung and stifle joint tissue samples. Each tissue pair originates from a single animal. For each tissue sample, 5 colonies were picked from the colony growth plate and sequenced.

Hill Lab Sequences at 250 bp:

B. 30 paired-read sets consisted of isolates sampled from 6 bovine tissue samples. The samples consist of 3 pairs of bovine lung and stifle joint tissue samples, with each pair originating from a single animal. For each tissue sample, 5 colonies were picked from the colony growth plate and sequenced.

C. 48 paired-read sets were sequenced from isolates grown from 24 bovine lung and stifle joint tissue sample pairs, with each pair of tissue samples originating from a single animal.

D. 3 paired-read sets were sequenced isolates *Mycoplasma arginini*, *Mycoplasma agalactiae*, and *Mycoplasma bovirhinis*.

E. 10 paired-read sets were *in vitro* mixes of 6 *M. bovis* isolates that had previously sequenced, and represented in F. and G. below.

Neogen Sequences at 150 bp:

F. 4 paired-read sets that were the same isolates as the isolates in the *in vitro* mix

Nation Research Council Sequences at 300 bp:

G. 2 paired-read sets that were the same isolates as the isolates in the *in vitro* mix

4.5 SepSIS Post-processing and Experiments

4.5.1 List of Mixes

A total of 194 *in silico* combinations of isolates and 10 *in vitro* combinations of isolates were created for the various experiments, in accordance with the goal in Section 3.2.2. The description for each experiment (Sections 4.5.4, 4.5.5, 4.5.6, 4.5.7, and 4.5.8) contains a list of which of the isolate mixture datasets were used in the experiment, because an experiment or verification may use more than one. This is represented by the identifying letter for each set of mixes (A-E). However, if the mixes were specifically created for a certain experiment, it is mentioned in the list of mixes that follows.

A. There were 10 *in vitro* isolate mixes created for experimentation with SepSIS. Numbering the isolates 1 to 6, the 10 combinations took the form of: 12, 13, 23, 123, 135, 345, 1234, 12345, 12346. These combinations were chosen to represent multiple sizes of combinations while also restricting the number of samples needed for their creation. These are the 10 read sets discussed in read group E above. Another 10 *in silico* mixes were created using previously-sequenced reads in the same mixing combinations as the isolates mixed *in vitro*. These reads are from groups F and G above.

B. A set of 33 *in silico* mixes was created to examine the ability of SepSIS to separate strain-specific sequences in mixes containing 2, 3, 4, and 5 isolates. These combinations were made of random isolates from read group C. The combinations take the form of 12, 23, 34, 45, 15, 123, 345, 135, 1234, 1235, and 12345.

C. A total of 110 *in silico* isolate mixes were created for the experiment to determine if multiple strains exist on the same plate. These 110 mixes were created from 10 sets of 5 isolates picked from the same colony plate. For each set of 5 isolates from a single colony plate, 11 combinations were created that varied in size from 2 – 5 isolates. Numbering the isolates 1 – 5, these combinations took the form of: 12, 23, 34, 45, 15, 123, 345, 135, 1234, 1235, and 12345. These isolates are represented in read groups A and B above.

D. A total of 29 pairs of lung and joint isolates were *in silico* mixed for the purpose of identifying any tropism-unique sequences for lung or stifle joint tropisms. All isolates from read group C, 2 pairs of lung and joint isolates from group A, and 3 pairs of lung and joint isolates from group B were used to create these mixes.

E. A set of 12 *in silico* combinations were created to test how the SepSIS pipeline reacts to contamination with other *Mycoplasma* species. These combinations consist of an *M. arginini* isolate, an *M. agalactiae* isolate, and *M. bovirhinis* isolate combined with 4 *M. bovis* isolates. In the read groups, these mixes were made from groups D and 4 random isolates from group C.

4.5.2 SepSIS Run Parameters for All Mixes and Output

All of the mixes listed above were run with identical settings through SepSIS, except for the *in vitro* mixes that could not be run using the validation method. The settings are presenting in Table 4.1. All mixes were run in all combinations of RUNMODE-s and SUBMODE-s. In addition, each coverage-based mode was run twice for each SUBMODE, with two different Max_Score_Value-s. The exact minimum and maximum score values were determined through preliminary testing runs. The Min_Score_Value for all ORGANIC_P runs was set to 10. CYCLIC ORGANIC_P runs were run with the parameter Max_Score_Value at 20 and then 30, while the ISOLATED and BOTH ORGANIC_P were run with parameters Max_Score_Value-s of 30 and then 35. Similarly, the Min_Score_Value for all ORGANIC_Z runs was set to -1.282, roughly equivalent

Table 4.1: The SepSIS run settings used on each mix of isolates. An iterative testing approach was used to assign the Min_Score_Value-s and Max_Score_Value-s. The SYNTH RUNMODE settings could not be used on the *in vitro* mixes due to the impossibility of labelling the reads with an isolate ID.

RUNMODE	SUBMODE	Min_Score_Value	Max_Score_Value
ORGANIC_P	CYCLIC	10	20
ORGANIC_P	CYCLIC	10	30
ORGANIC_P	ISOLATED	10	30
ORGANIC_P	ISOLATED	10	35
ORGANIC_P	BOTH	10	30
ORGANIC_P	BOTH	10	35
ORGANIC_Z	CYCLIC	-1.282	-0.842
ORGANIC_Z	CYCLIC	-1.282	-0.524
ORGANIC_Z	ISOLATED	-1.282	-0.524
ORGANIC_Z	ISOLATED	-1.282	-0.385
ORGANIC_Z	BOTH	-1.282	-0.524
ORGANIC_Z	BOTH	-1.282	-0.385
SYNTH	CYCLIC	1	1
SYNTH	ISOLATED	1	1
SYNTH	BOTH	1	1

to the 10th percentile. The CYCLIC ORGANIC_Z runs were run with the parameter Max_Score_Value set to -0.842 and then -0.524 and the ISOLATED and BOTH ORGANIC_Z runs were run with the parameter Max_Score_Value set to -0.524 and then -0.385. These values were determined through iterative testing that is discussed in Section 6.2.4.

The ISOLATED and BOTH SUBMODE-s have a tipping point of approxamately the 35th to 40th percentile or a Z-Score of -0.385 to -0.2533 where the Max_Score_Value ceases to function as an effective threshold. This is because exponentially large numbers of sequences are classified as strain-specific, decreasing accuracy and increasing runtimes. However, if the upper selected threshold is lowered below a particular setting, generally around the 33rd percentile mark, there ceases to be output. When the mixes were run in the SYNTH method, the Max_Score_Value and Min_Score_Value were set to 1 across all SUBMODE-s to ensure each node was completely strain-specific.

4.5.3 Data Post-Processing

The post-processing steps are represented in Figure 4.10. Post-processing begins with the validation of the strain-independent subsequences on one or both ends of a sequence produced by SepSIS. This is step A in Figure 4.10. Within each strain-independent subsequence, the minimum of the two following values is

determined from the subsequence: the length of the subsequence, or a length of 1500 bp closest to the strain-specific subsequences. This is performed using the information found in the file name (it contains an indicator of which ends have strain-independent subsequences) and the subsequence headers (they contain subsequence length and position). The cutoff of 1500 bp was chosen for two reasons: to prevent extremely long nodes from strongly affecting any later nucleotide-nucleotide comparison steps because the length of the sequences are used in the equations of those steps, and to allow for a section of the nucleotides close to the strain-specific sections of the sequence to be analyzed in the case of misassembly of extremely long strain-independent subsequences (such as a subsequence upward of 20,000 nucleotides).

The extracted strain-independent subsequences are BLASTN-ed against the 11 complete *M. bovis* genomes available on the National Center for Biotechnology Information (NCBI) website [9, 44]. Sequences that have greater than or equal to 94% ANI along the entire strain-independent subsequence are deemed true *M. bovis* subsequences, in accordance with the Konstantinidis' research into species and strains in Section 2.2 [19]. Any strain-specific subsequences that are not attached to at least one verified (passed the 94% ANI threshold) strain-independent subsequence are removed before further analysis. Sequences that started this step with strain-independent subsequences at both ends, but possess only one verified end are moved to the appropriate output file (the "FrontEnds" or "BackEnds" file as mention in Section 4.3.2). Note that this method removes any strain-independent subsequences unique to a particular isolate if the subsequences do not exist within an existent reference assembly. This step could include the removal of correctly assembled sequences that do not exist within the reference strains. However, the value in this step is that it ensures that the strain-independent subsequences are not misassemblies. Therefore, any labratory primers derived from the strain-independent subsequences would not be a waste of resources to develop (a goal in Section 3.1.4).

During the growth of *M. bovis* in our lab, contamination with *Stenotrophomonas maltophilia* on the culture plates occurred on occasion. Some of the sequenced *M. bovis* isolates were found to be at least partially contaminated with *S. maltophilia*. To combat potentially contaminating *S. maltophilia* subsequences in the SepSIS output, the SepSIS output sequences are compared to an *S. maltophilia* reference genome. This is step B in Figure 4.10. The strain-specific section of each sequence is extracted from each of the output files, and those subsequences are BLASTN-ed against the K279a strain of *S. maltophilia*. If the strain-specific section of a sequence is found to have a greater than a 94% ANI (along the entire subsequence) to *S. maltophilia* reference genome, it is then BLASTN-ed against the 11 existing completed *M. bovis* reference genomes available from NCBI [44]. If the subsequence also has greater than a 94% ANI (along the entire subsequence) to an *M. bovis* reference genome, it is deemed that it is not contaminated. This is because a subsequence with high ANI to both genomes is likely to be a shared subsequence between the two species, rather than a contaminant. A subsequence is deemed a contaminant and removed if it has a greater than a 94% ANI to *S. maltophilia*, but not to *M. bovis*. The results of this process are discussed alongside the results of purposeful contamination in the SepSIS pipeline in Section 5.4.

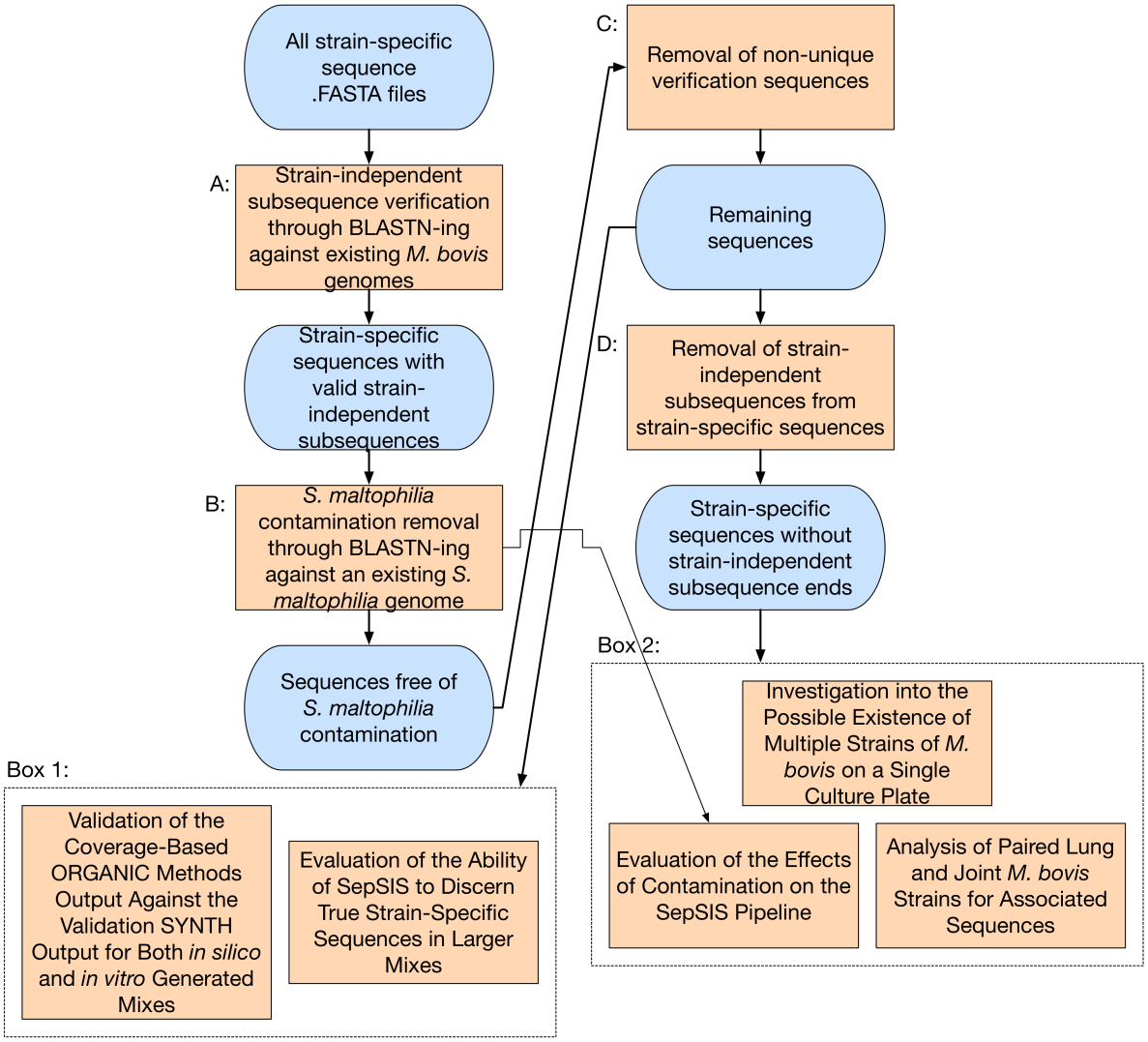

Figure 4.10: The post-processing steps for the output from SepSIS. The shared steps are labelled steps A to D. Box 1 contains the validation experiments for the functionality of SepSIS, while Box 2 contains the experiments that use SepSIS to investigate data.

It was discovered late in processing that the sequences produced by the SYNTH RUNMODE of the SepSIS are not always specific to a particular strain in the case of SPAdes creating multiple assemblies of a single sequence. This is much more likely to happen with a larger number of mixed isolates due to the limitations of the SPAdes assembler. These limitations are discussed in Section 6.3.1. Therefore, the next step is to perform pairwise comparisons against all other sequences within each output file. In Figure 4.10, this is step C. If one sequence matches to another sequence at 100% ANI across the first sequence's entire length, it is added to a removal list. If the sequence's are the same length, both sequences are added to the removal list. All sequences in the removal list are removed from further analysis. The remaining non-duplicated sequences are the final output sequences of the pipeline.

As an additional step, some experiments require analysis of only the strain-specific portion of the sequences produced by runs with the validation settings. These experiments are described in Sections 4.5.6, 4.5.7, and 4.5.8. For these experiments, the strain-specific portion of each sequence from each file is extracted, resulting in a list of strain-specific sequences without a strain-independent end or ends. This is represented as step D in Figure 4.10. As a small additional step for easier processing, the validation sequence files are copied and split into different files containing reads from only one isolate in the mix. This is possible due to the annotation at the beginning of each sequence.

4.5.4 Validation of the Coverage-Based ORGANIC Modes Output Against the Validation SYNTH Output for Both *in silico* and *in vitro* Generated Mixes

The validations of the coverage-based ORGANIC_Z and ORGANIC_P modes are performed by comparing the output after post-processing against that of the output from the validation SYNTH method after post-processing for every *in silico* combination, in fulfillment of the first goal discussed in Section 3.3.1. In total, 7 sets of mixes were compared in this manner. These sets are referred to by name in later portions of this thesis (e.g.: the set of *in silico* mixes or the set of paired isolates mixes). A description of these sets and the method used to compare the sets follows.

- The Set of *in silico* Mixes: These mixes are discussed in group A of the List of Mixes in Section 4.5.1 and are made up of the previously sequenced isolates. These mixes act as the synthetic ground truth for comparison with the *in vitro* mixed isolates because *in vitro* mixed reads do not have meta-information on which reads belong to which isolate. Additionally, the comparison of the coverage-based and verification methods results for this set of mixes is a baseline to assess how well the coverage-based methods of SepSIS can assess the presence of strain-specific sequences in randomly mixed isolates.

- The Set of *in vitro* Mixes: These mixes consist of the *in vitro* mixed isolates in group A of the List of Mixes. The comparison of the *in vitro* mixes with The Set of *in silico* Mixes allows for an assessment of how well the coverage-based modes of SepSIS can identify strain-specific sequences in real world mixes

of isolates.

- The Set of Paired Isolates Mixes: These mixes consist entirely of the 29 mixes from group D found in the List of Mixes. The objective of analyzing this mix is to determine if the coverage-based modes of SepSIS can isolate and identify strain-specific sequences from highly similar isolates.

- The Sets of Large Mixes Containing 2, 3, 4, or 5 Isolates: These mixes are represented in group B in the List of Mixes. These mixes were created to examine the similarity of coverage-based modes results to the validation method results when mixes containing 3, 4 or 5 randomly-chosen isolates are processed with SepSIS. The mixes are split into analysis sets based on the number of isolates in the set.

Each individual mix in each group above was run through SepSIS a total of 15 times. Of these runs, 12 were coverage-based modes, and 3 were verification method runs. The selected options for the coverage-based modes are described in Section 4.5.2. Each coverage-based run had all output sequences compared against the output for validation run with parameters SYNTH, Min_Score_Value=1, Max_Score_Value=1, and the relevant SUBMODE.

The methodology taken to examine and validate the coverage-based datasets follows and is also presented in Figure 4.11. Each set of 15 runs of SepSIS is performed on a single *in vitro* mix of isolates meaning that all 15 runs use an identical assembly graph as input. The identical assembly graph allows for the comparison of the sequences between different runs using sequence headers instead of nucleotide sequence. This method of comparison is possible because if the two SepSIS runs with different parameters output sequences with the same header name, the sequences are also identical. This is step A in Figure 4.11.

During the validation of the *in vitro* coverage-based modes the *in silico* SYNTH output for the 10 *in vitro* mixes are compared to the 10 *in silico* mixes of the same isolates using BLASTN instead of header name comparisons. Note that only the 12 coverage-based runs were performed for the *in silico* mixes due to a lack of meta-information necessary for the validation method runs. In Figure 4.11, this is the path leading to step B. These comparisons are performed using a BLASTN cutoff of 94% ANI and an additional threshold for length between the two sequences being compared. If the *in vitro* query sequence was not within 98% or 102% of the length of the *in silico* target sequence it was deemed to not be the same strain-specific sequence. This was performed to account for errors that may have occurred during sequencing or assembly. The BLASTN cutoff was chosen after multiple calibrating iterations and is discussed in Section 6.2.1.

Both of the above comparison methods produce the number of strain-specific sequences that the coverage-based modes positively identified (true positive), sequences identified as strain-specific but are not (false positive), and unidentified strain-specific sequences (false negative). These values were totalled for each individual mix. A true negative value is unavailable for the data because that value would require counting of all possible strain-independent sequences in the assembly graph. Finding all such sequences would require its own algorithm and would involve an exponentially increasing number of possible strain-independent sequences. This would result in a massive number of true negative sequences that would not produce informative statistics.

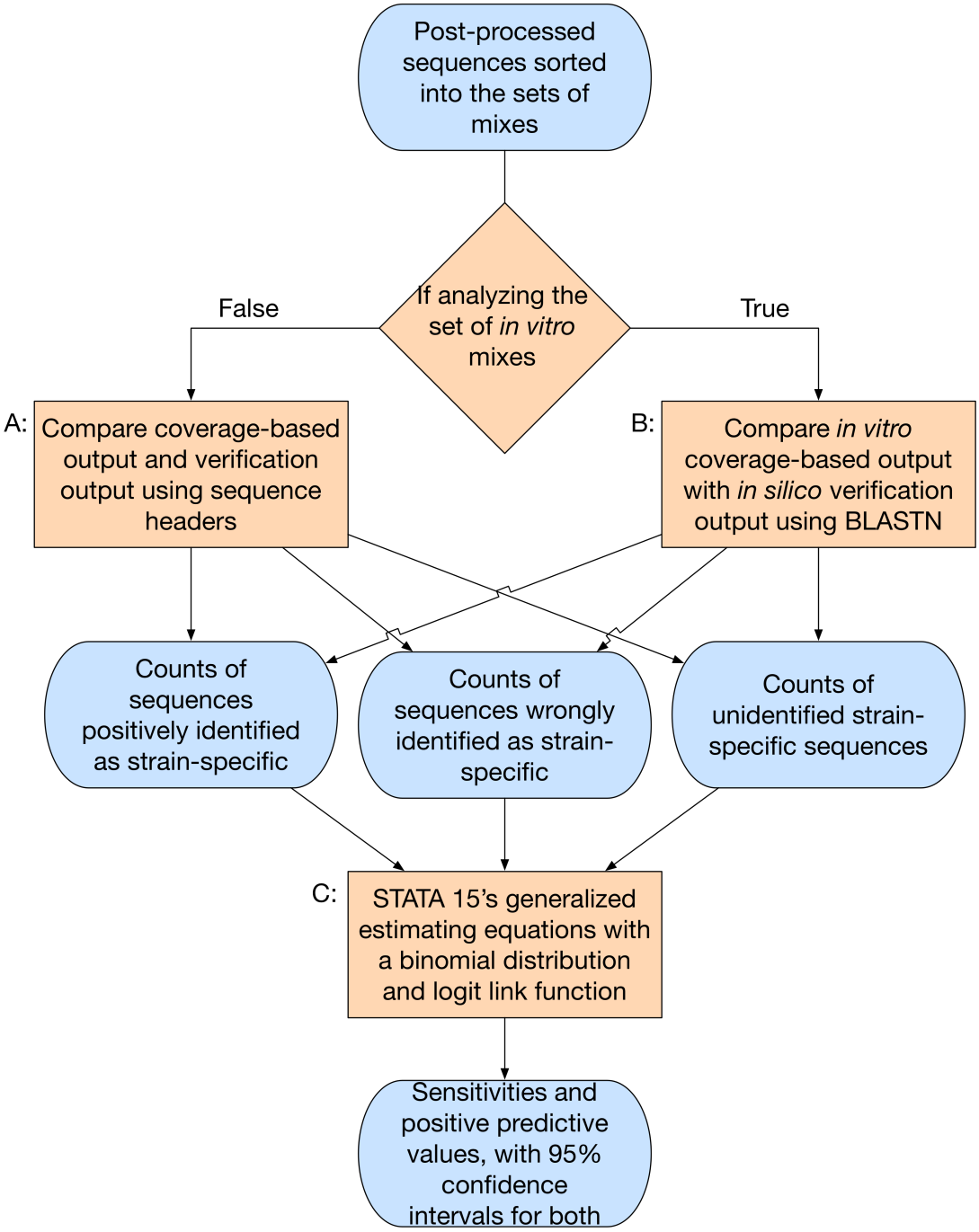

Figure 4.11: The steps taken during the validation of the coverage-based ORGANIC modes output against the validation SYNTH output. Different steps are taken depending on the dataset being analyzed. The path through step A is taken if an *in silico* dataset is being evaluated and the path through step B is taken if an *in vitro* dataset is being evaluated. Both paths produce counts of sequences that are used as input to STATA 15 to generate the sensitivity and PPV values with 95% confidence intervals.

For each set of mixes, the sensitivity, the positive predictive value (PPV), and 95% confidence intervals for the sensitivity and PPV are calculated using STATA 15 [46]. This is presented as step C in Figure 4.11. The sensitivity is a measure of the probability that a truly strain-specific sequence will be correctly identified by the coverage-based mode and is calculated using the equation `(true positives/(true positives + false negatives))`. The PPV is the probability that a sequence identified as strain-specific by the coverage-based mode is truly strain specific. PPV is calculated using the equation `(true positives/(true positives + false positives))`.

In STATA, the PPVs and sensitivities are generated using generalized estimating equations (GEE) with a binomial distribution and logit link function (which is the equivalent of logistic regression). GEE was used due to the possibility of there being correlation between the observed outcomes. This correlation can occur due to the same isolates being used to create multiple mixes in a single set of mixes. An example of this occurs in the dataset the set of large mixes containing 2 isolates. Each individual isolate is present in two mixes in the dataset. This method generated 95% confidence intervals for PPV and sensitivity. The logistic regression results allow an estimation of the sensitivity and the PPV producing the results presented in Section 5.1.

4.5.5 Evaluation of the Ability of SepSIS to Discern True Strain-Specific Sequences in Larger Mixes

As discussed in Section 3.3.2, the secondary goal of the analysis of SepSIS output was to investigate how SepSIS handles larger mixes of isolates. The steps of this evaluation are presented in Figure 4.12 and referenced later in this section. The data is group B from Section 4.5.1, also named The Set of Large Mixes in Section 4.5.4. These mixes are represented in Figure 4.12 at A. This inquiry attempts to learn if strain-specific sequences identified by the verification method were falsely described as strain-specific and how the quantity of falsely described sequences changes with greater numbers of isolates.

As larger numbers of isolates are added to an *in silico* mix, the ability of SPAdes to create a coherent assembly decreases. This can be seen both in the number of contigs produced by SPAdes and in the increasing length of all contigs in the output from SPAdes. Additionally, a large number of highly similar sequences in the output assembly graph may create difficulties for minimap2 to map reads to the assembly graph subsequences. Because SepSIS relies upon output from those tools to function, further investigation into the verification output is necessary.

To investigate the effect of the presence and quantity of sequences falsely described as strain-specific, each isolate's strain-specific sequences are extracted from the .FASTA files of every mix containing it. This is represented by B in Figure 4.12. The isolate ID at the beginning of the sequence headers in the SepSIS output .FASTA files and the mix name are used to sort these these sequences into sets. The strain-specific sequences for each isolate from a single mix are BLASTN-ed against that same isolate's strain-specific sequences in every other mix. The results are thresholded using an ANI of 99% and a length requirement that the query

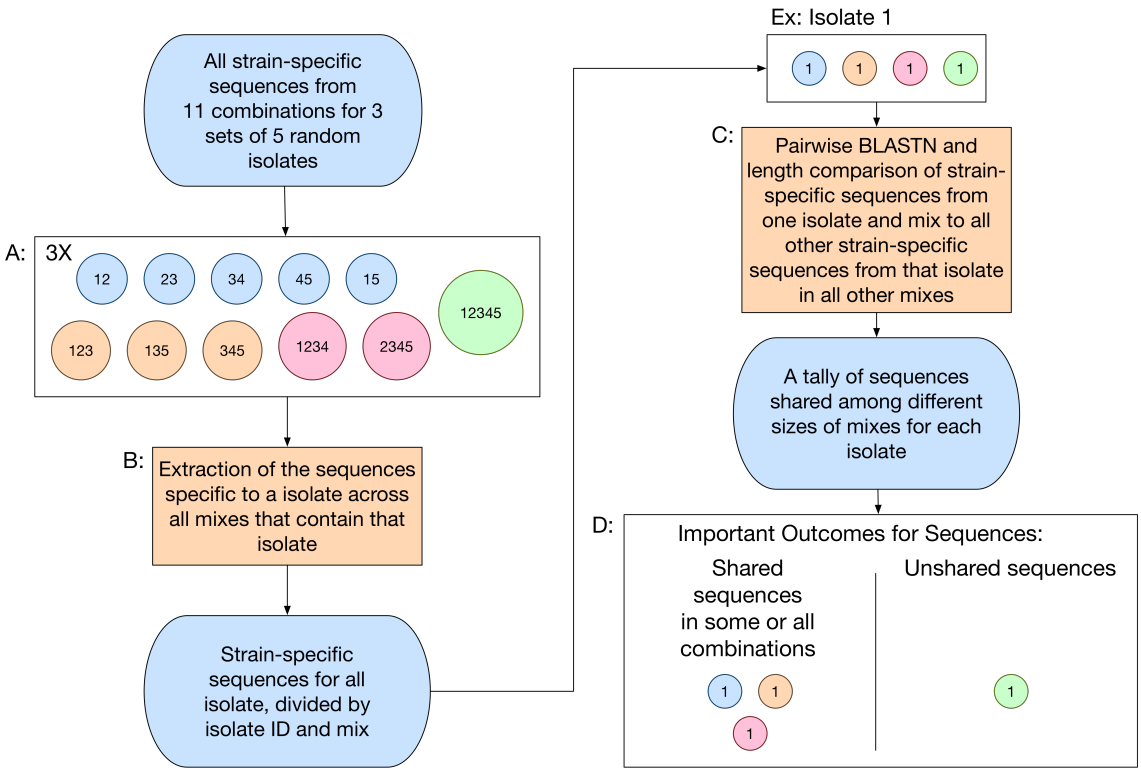

Figure 4.12: The steps taken during the evaluation of the ability of SepSIS to discern true strain-specific sequences in larger mixes. The composition of the mixes are shown in the bubbles in A, with each number (eg. "1", "2", "5") representing an isolate and the combination of size and colour of the bubble representing a mix size. Mixes of size 2 are blue and the smallest size circle, mixes of size 3 are orange and the 2nd smallest size, and so forth. The results from the extraction of sequences specific to isolate 1 from among the different mix sizes are shown after step B. In D, the circles shown represent the possibilities for the strain-specific sequences from strain 1: either the strain-specific sequences are shared among mix sizes, or they are unshared.

sequence is within the range of 98% to 102% of the length of the target sequence. Sequences within these parameters are classified as shared among multiple mixtures. Sequences that fall outside the parameters are unshared between mixes and potentially erroneous. In Figure 4.12, this is step C. For each size group for mixes, the number of sequences shared across multiple mixes and the number of unshared sequences are tallied. From these numbers, the number of strain-specific sequences that are unshared among mix sizes can be identified and presented in Section 5.2.

The concept behind this investigation is to determine if there exist sequences identified as strain-specific in mixes containing 3, 4, or 5 isolates that do not appear in other mix sizes. For instance, if a sequence identified as strain-specific for isolate 1 appears in mixes including that isolate of sizes 2, 3, and 4 then that sequence is logically likely to be a properly assembled, truly strain-specific sequence. However, if a sequence only appears in a mix of 5 isolates, then that sequence is likely to be a false assembly, or falsely flagged as strain-specific. This example is represented in Figure 4.12 as D. The proportion of unshared to shared sequences represents the proportion of wrongly-assembled sequences or sequences falsely identified as strain-specific.

4.5.6 Investigation into the Possible Existence of Multiple Strains of *M. bovis* on a Single Culture Plate

The first biological investigation conducted was to determine if multiple strains of *M. bovis* grown from a single biological sample exist on a single culture plate at one time, in accordance with the objective in Section 3.4.1. Note that this experiment and all further experiments use verification method output sequences with strain-independent subsequences removed, shown in step D and Box 2 of Figure 4.10. However, the metadata for the strain-independent subsequences still exists within the sequence headers, allowing for the strain-independent subsequences to be easily restored from the raw SepSIS output files if needed. The steps of this investigation are presented in Figure 4.13.

A total of 110 mixes of 50 *M. bovis* isolates are used for this task, represented by group C in the list of mixes in Section 4.5.1, and at A in Figure 4.13. This investigation shares many steps with Section 4.5.5 and is presented in Figure 4.13. For each isolate in the set of mixes, that isolate's strain-specific sequences are extracted from the .FASTA files of every mix containing it. This is done using the isolate ID appended to each .FASTA sequence header by SepSIS and is step B in Figure 4.13. For each individual isolate, the strain-specific sequences for that isolate from each mix is BLASTN-ed against that same strain's strain-specific sequences in every other mix. A cutoff of 99% ANI and a requirement that the query sequence be within 98% to 102% of the length of the target sequence is applied. This is step C in Figure 4.13. It is at this stage that this methodology differs from Section 4.5.5. The query sequences that pass this cutoff and requirement are sorted based on the size of the mix and the isolate ID they originate from. A tally of sequences shared among different mix sizes is produced for each isolate.

The difference in this Section from Section 4.5.5 is that this analysis focuses on the presence of singular shared strain-specific sequences across mix sizes for each single isolate, rather than the proportion of shared

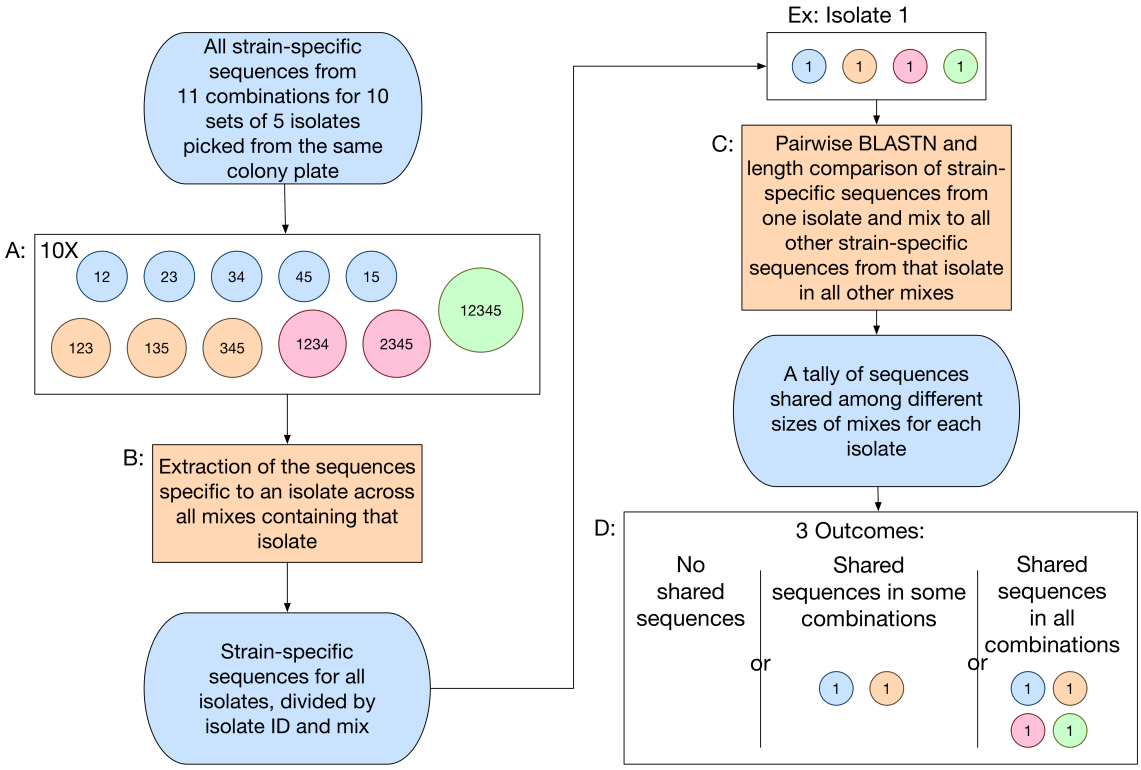

Figure 4.13: The steps taken during the investigation into the possible existence of multiple strains of *M. bovis* on a single culture plate. The composition of the mixes are shown in the bubbles in A, with each number (eg. "1", "2", "5") representing an isolate and the combination of size and colour of the bubble representing a mix size. Mixes of size 2 are blue and the smallest size circle, mixes of size 3 are orange and the 2nd smallest size, and so forth. The results from the extraction of sequences specific to isolate 1 from among the different mix sizes are shown after step B. step D shows the outcomes for each individual isolate, using isolate 1 as an example. An isolate will have only one of the following: no strain-specific sequences shared among mix sizes, strain-specific sequences shared among smaller mix sizes, or strain-specific sequences shared among all mix sizes. The "or" in step D is exclusive. Additionally, the "Shared sequences in some combinations" outcome is expressing that isolate 1 has strain-specific sequences shared in the SepSIS results from mixes of size 1 and size 2.

to unshared sequences. In this experiment it is expected that if every isolate on a single culture plate is an identical strain, then minimal or no strain-specific sequences would be identified that are shared across all sizes of mixes. If a isolate does have a strain-specific sequence present across multiple mix sizes then it can be declared that the isolate is a genomically unique strain despite sharing a culture plate with other strains of the same species.

There are 3 possible trends that occur across the strain-specific sequences. These cases are also presented in 4.13 as box D.

- Case 1: For a single isolate, there are no strain-specific sequences that are shared among mixes or there are very few strain-specific sequences shared among smaller mixes. This could imply that the other isolates in the mix are identical to the strain of the isolate being evaluated, or that the sequences of the isolate being analyzed are a proper subset of another isolate.

- Case 2: For a single isolate, at least one single strain-specific sequence is shared among multiple mixes of 2, 3, or 4 isolates. This suggests that the isolate may have some strain-specific sequences, but the some sequences identified as strain-specific are shared with at least one other isolate on the plate.

- Case 3: For a single isolate, at least one single strain-specific sequence is shared among all sizes of mixes. This suggests that the isolate possesses strain-specific sequences that are unique relative to all other isolates from the plate, and that the single isolate is a unique strain relative to the other strains on the plate.

While the results for this experiment do not provide the strain-specific sequences for each isolate, the purpose of this experiment is to investigate whether isolates of a single species on a plate are clonal. The results are discussed in Section 5.3.

4.5.7 Evaluation of the Effects of Contamination on the SepSIS Pipeline

This evaluation was performed to determine if the processing steps designed to remove contamination of *M. bovis* data are successful. This is discussed in the objectives Section 3.4.2, and uses the 12 *in silico* mixes from group E in Section 4.5.1 containing 4 strains of *M. bovis* combined pairwise with a strain of *M. bovirhinis*, *M. arginini*, or *M. agalactiae*. This evaluation is performed by separating the strain-specific sequences into sets based on the strain ID, giving a set of *M. bovis* strain-specific sequences and non-*M. bovis* strain-specific sequences for each mixture. These sequences were then BLASTN-ed against the PG45 *M. bovis* genome, and the relevant reference genome for the non-*bovis* species at an ANI of 94%. No further steps were required, as discussed in 5.4.

Note that there are two steps in the post-processing stages in Section 4.5.3 that assist in removing contaminant sequences from any SepSIS output. The first is the validation of the strain-independent subsequences as *M. bovis* sequences (step A in Figure 4.10). The remaining sequences tagged as strain-specific in the

contamination mixes were BLASTN-ed against the *M. bovis* genome and the primary reference genome of the contaminant species to determine if any strain-specific sequences from contaminant species are present in the SepSIS output. The second contaminant removal stage in post-processing is to address possible and unintentional *S. maltophilia* contamination (step B in Figure 4.10). This step produces a list of strain-specific sequences that are likely to be contaminated with *S. maltophilia*, and removes them from further post-processing. These sequences and the strains and mixes they belong to are discussed in Section 5.4.

4.5.8 Analysis of Paired Lung and Joint *M. bovis* Isolates for Tropism-Specific Sequences

The final experiment is to determine if there exist strain-specific sequences in *M. bovis* that are specific to stifle-joint-tissue tropism or lung-tissue tropism using 29 lung and joint *M. bovis* isolate pairs, in accordance with the goal in Section 3.4.3. The 29 pairs of isolates are discussed in group D of Section 4.5.1. Figure 4.14 presents the steps to isolate the tropism-specific sequences. The methodology for this experiment is as follows. First, the strain-specific sequences from all lung isolates are individually BLASTN-ed as query sequences against all the strain-specific sequences from the joint tropisms. All strain-specific sequences from the joint isolates are also individually BLASTN-ed as query against the strain-specific sequences from all lung isolates. These are respectively steps A1 and A2 in Figure 4.14. If a sequence from one tropism matches with a sequence from the opposing tropism at 100% ANI, that sequence is removed from further processing. It is removed because that sequence is not tropism-specific due to matching a sequence from the opposing tropism. The rate of 100% ANI was chosen to ensure that sequences with small single SNP differences are still counted as strain-specific.

The remaining sequences from each tropism are then BLASTN-ed against all other sequences from the same tropism to find common strain-specific sequences across that tropism. These are steps B1 and B2 in Figure 4.14. Because BLASTN produces pairwise relations, the headers of sequences matching at 100% ANI are placed into a list of pairs. These pairs are then compared to produce a list of sequence headers from different tropisms that all represent the same sequence. The number of isolates possessing a single common sequence and the number of sequences shared among isolates are reported in Section 5.5.

The most common sequences for both tropisms are BLASTN-ed against a locally downloaded version of the NCBI nr database to identify the gene or genes that the sequences map to [44]. The database was downloaded on December 10th, 2019. The top 10 matches presented by BLASTN for each query were assessed by hand, and the most common matches are presented in Section 5.5.

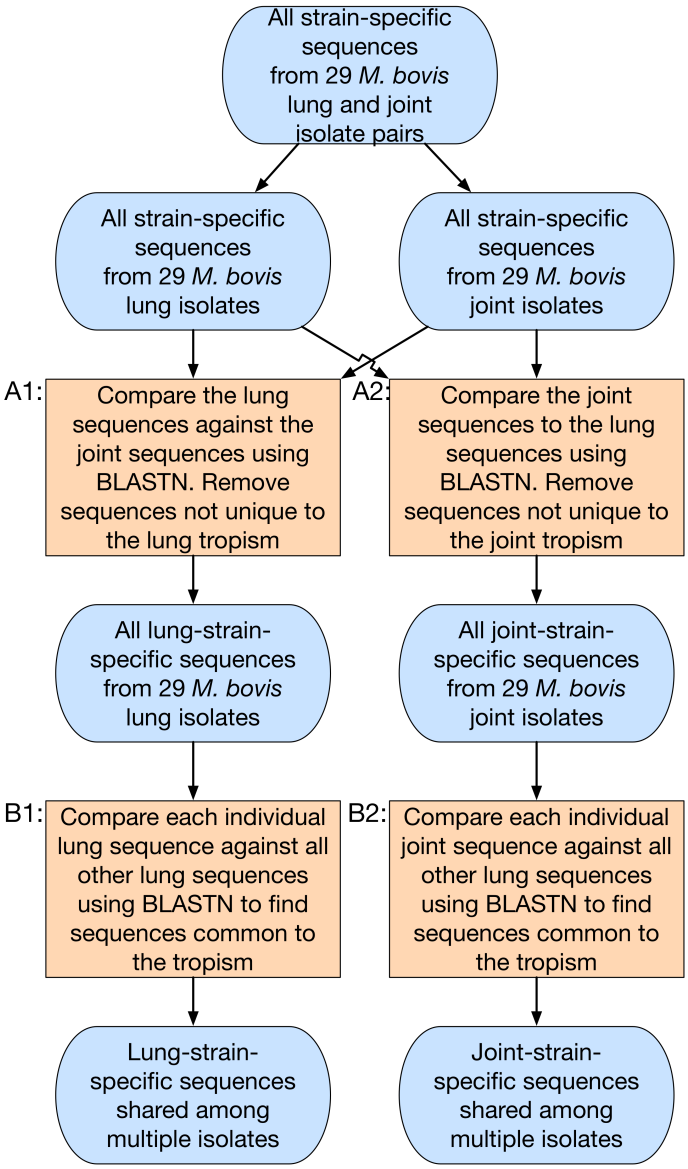

Figure 4.14: The steps taken during the analysis of paired lung and joint *M. bovis* strains for tropism-specific sequences. The strain-specific sequences for the lung strains and joint strains are separated into groups and processed separately, but with nearly identical methodologies. They differ at each step, exchanging 'lung' and 'joint' where appropriate.

5 Results

This chapter contains 5 sections describing the results from validations and analyses described in Sections 4.5.4 through 4.5.8. Section 5.1 contains the results evaluating the coverage-based methods of SepSIS. Section 5.2 is an evaluation of the validation method's ability to handle mixes with 3 to 5 strains. Section 5.3 is an investigation into the existence of multiple strains of *M. bovis* on a single culture plate. Section 5.4 discusses the results from post-processing steps taken to remove contamination and the results of purposeful synthetic contamination of some of the mixes. Section 5.5 presents the results from the investigation into sequences that are phenotype-specific for the paired lung and stifle joint mixes.

5.1 Results of the Validation of the Coverage-Based ORGANIC Modes Output Against the Validation SYNTH Output for Both *in silico* and *in vitro* Generated Mixes

As previously described in Section 4.5.4 and in fulfillment of the goals in Section 3.3.1, the output from each run of the coverage-based methods ORGANIC_Z and ORGANIC_P were compared against the output from the validation method SYNTH to assess the ability of the coverage-based approaches to produce output similar to the meta-information-based validation method. Each run of the coverage-based method is referred to by the input parameters used in the format: RUNMODE SUBMODE [Min_Score_Value, Max_Score_Value]. The results for this subsection are presented in histograms (Figures 5.1, 5.2, 5.3, 5.4, 5.5, and 5.6) matching 6 of the 7 sets of mixes discussed in Section 4.5.4. A histogram is not presented for the set of *in vitro* mixes due to a lack of positive results, as discussed in that dataset's subsection. The histograms contain the sensitivity and PPV presented as a probability for each comparison of a coverage-based mode with previously described parameters. These parameters are represented on the x-axis as the RUNMODE, SUBMODE, and the Min_Score_Value and Max_Score_Value for the run in square brackets. The data used to create these figures is presented in Appendix A, Tables A.3, A.4, A.5, A.6, A.7, A.8, and A.9. The raw counts of strain-specific sequences used to create the values in these tables are presented in Appendix A, Tables A.10, A.11, A.12, A.13, A.14, A.15, and A.16

5.1.1 The Set of *in silico* Mixes

The results of the validation of the coverage-based runs of the set of *in silico* mixes are presented in Figure 5.1, with associated data in Tables A.3 and A.10. In this figure, it is shown that the coverage-based modes were only partially successful in modelling the verification dataset. Note that this analysis did not separate the values to calculate sensitivity and PPV based on mix size, meaning that mixes of 2 through 5 isolates are all taken into consideration as a group to calculate the sensitivity and PPV. In this subsection Figure 5.1 is described in depth. The description serves as a basis for understanding the successfulness of other datasets run through SepSIS.

Starting from left to right in Figure 5.1, the ORGANIC_P CYCLIC [10,20] run shows a roughly even PPV and sensitivity at approximately 0.12. In the ORGANIC_P CYCLIC [10,30] run, a spike in sensitivity can be seen up to 0.27 with little change in PPV. This is because the wider range in the Min_Score_Value-s and Max_Score_Value-s under CYCLIC provides a larger pool of sequences that may be correct. However, the unchanging PPV implies that the likelihood of an individual sequence being truly strain-specific does not change.

The ORGANIC_Z CYCLIC [-1.282,-0.842] runs failed entirely. Next, ORGANIC_Z CYCLIC [-1.282,-0.524] had a relatively high PPV, but a low sensitivity. Note that STATA was unable to calculate the 95% confidence interval for the PPV. As can be seen in the four leftmost run conditions of the histogram, the Z-Score-based mode has much lower sensitivity than the percentile-based mode. This is due to a smaller number of sequences predicted as strain-specific by the mode. The Z-Score-based mode also has a slightly higher PPV than the ORGANIC_P mode. This shows that the predicted sequences have a slightly higher likelihood of being truly strain-specific.

The ISOLATED runs for both ORGANIC_P and ORGANIC_Z were both poor, but in different ways. The ISOLATED ORGANIC_P runs had very similar PPVs and sensitivities to the CYCLIC runs, except that the 95% confidence intervals were massive. Therefore, the reported values were somewhat unreliable. The ORGANIC_Z runs produced no results at all. This is unsurprising, because poorer results were expected of the ISOLATED mode due to the low quality sequences comprising the ISCCs. As discussed in Section 4.1.2, the ISCCs are isolated, fragmented sections of the assembly with low coverage.

The BOTH runs were executed on the whole assembly graph produced by SPAdes. The ORGANIC_P BOTH runs had relatively high sensitivities, with ORGANIC_P BOTH [-1.282,-0.524] and ORGANIC_P BOTH [-1.282,-0.385] having values of approximately 0.25 and 0.31, and PPVs at 0.11 and 0.11 respectively. The ORGANIC_Z BOTH runs were much worse than the ORGANIC_P BOTH runs. ORGANIC_Z BOTH [-1.282,-0.524] produced a sensitivity near 0 due to the SepSIS run producing few strain-specific sequences. The ORGANIC_Z BOTH [-1.282,-0.385] run had a much higher sensitivity at 0.17, but with 95% confidence interval range of approximately 0.20. Additionally, the PPV is lower than most other runs at 0.06.

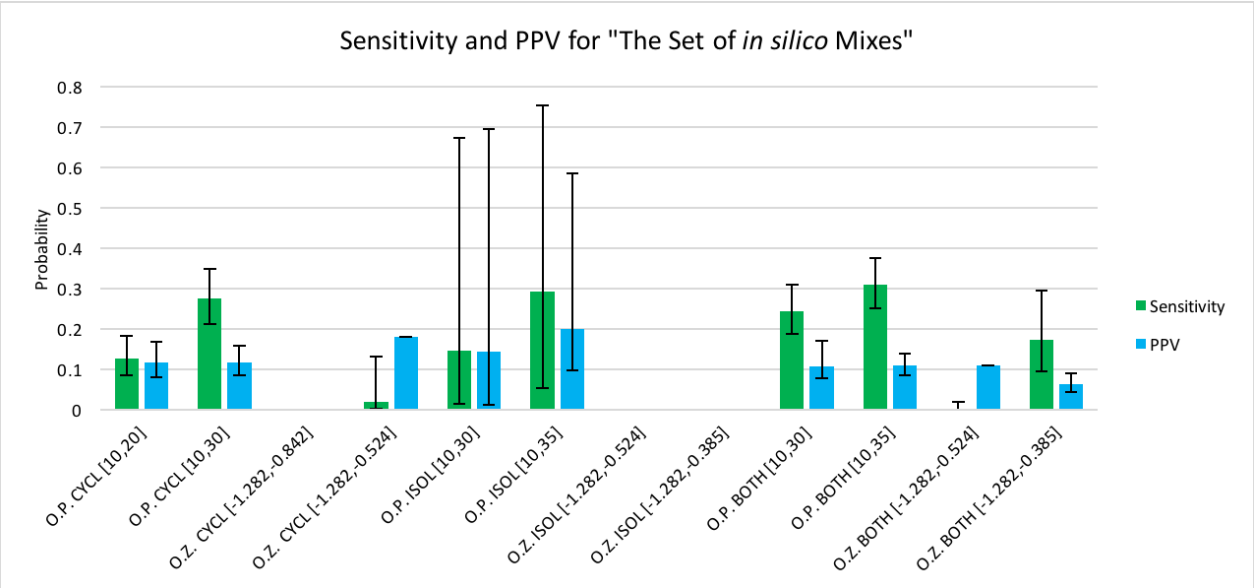

Figure 5.1: The sensitivity and the positive predictive value for the set of *in silico* mixes. The probability as determined by STATA 15 is represented on the y-axis, and an abbreviated set of each of the SepSIS run conditions for the group of mixtures is represented in the x-axis. The error bars represent the 95% confidence interval for the results. The run conditions are abbreviated as follows: O.P. = ORGANIC_P RUNMODE, O.Z. = ORGANIC_Z RUNMODE, CYCL = CYCLIC SUBMODE, ISOL = ISOLATED SUBMODE, BOTH = BOTH SUBMODE, and the final conditions are the Min_Score_Value and Max_Score_Value represented as [Min_Score_Value, Max_Score_Value].

5.1.2 The Set of *in vitro* Mixes

The results from the coverage-based method runs of SepSIS on the set of *in vitro* mixes did not match successfully against the results from the set of *in silico* mixes. In no case did any sequences identified as strain-specific in the *in vitro* dataset match against the *in silico* modelled dataset. Additionally, very few sequences were identified as strain-specific in the *in vitro* dataset. Therefore, there is no modelling histogram, though the raw numbers are shown in Tables A.4 and A.11.

5.1.3 The Set of Paired Isolates Mixes

The results from the runs on the set of paired isolates mixes are shown in Figure 5.2, with raw numbers in Tables A.5 and A.12. As seen in Figure 5.2, the coverage-based modes of SepSIS did not function as well on this dataset as on the *in silico* dataset. The CYCLIC and BOTH modes performed very poorly, with all but one PPV under 0.10 and all sensitivities under 0.10. This could be due to the high similarity of these isolates affecting assembly and coverage ratios in such a way that decreases the effectiveness of SepSIS. In any case, the results from mixing highly similar isolates are much worse than the set of *in silico* mixes constructed from random isolates.

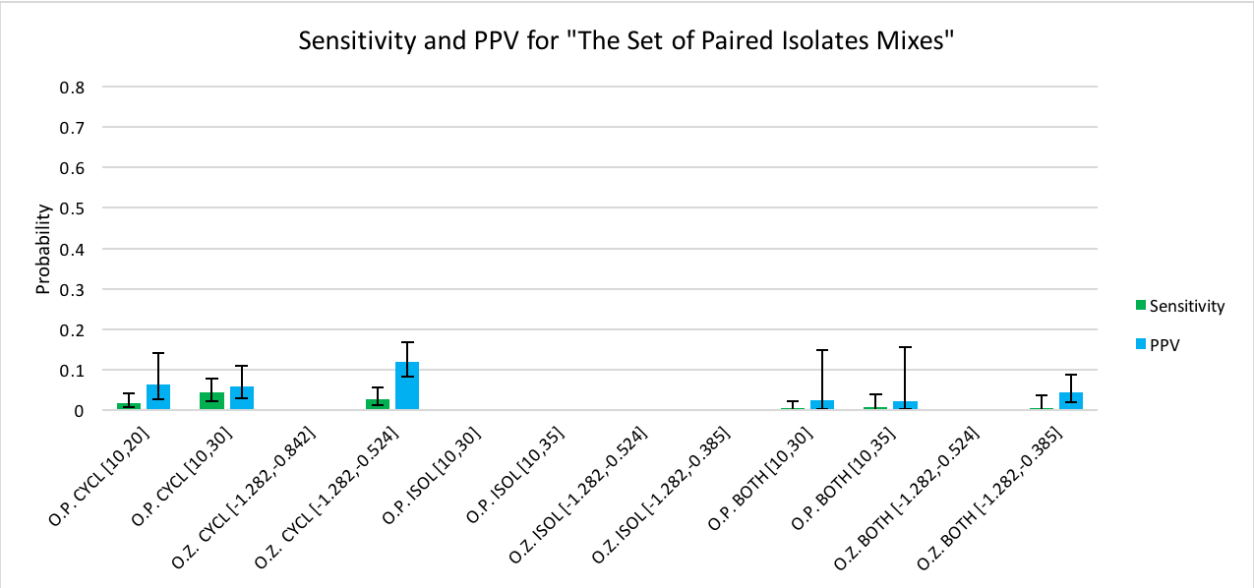

Figure 5.2: The probabilities of the sensitivity and the positive predictive value for the set of paired isolates mixes. The probability as determined by STATA 15 is represented on the y-axis, and an abbreviated set of each of the SepSIS run conditions for the group of mixtures is represented in the x-axis. The error bars represent the 95% confidence intervals for the results. The run conditions are abbreviated as follows: O.P. = ORGANIC_P RUNMODE, O.Z. = ORGANIC_Z RUNMODE, CYCL = CYCLIC SUBMODE, ISOL = ISOLATED SUBMODE, BOTH = BOTH SUBMODE, and the final conditions are the Min_Score_Value and Max_Score_Value represented as [Min_Score_Value, Max_Score_Value].

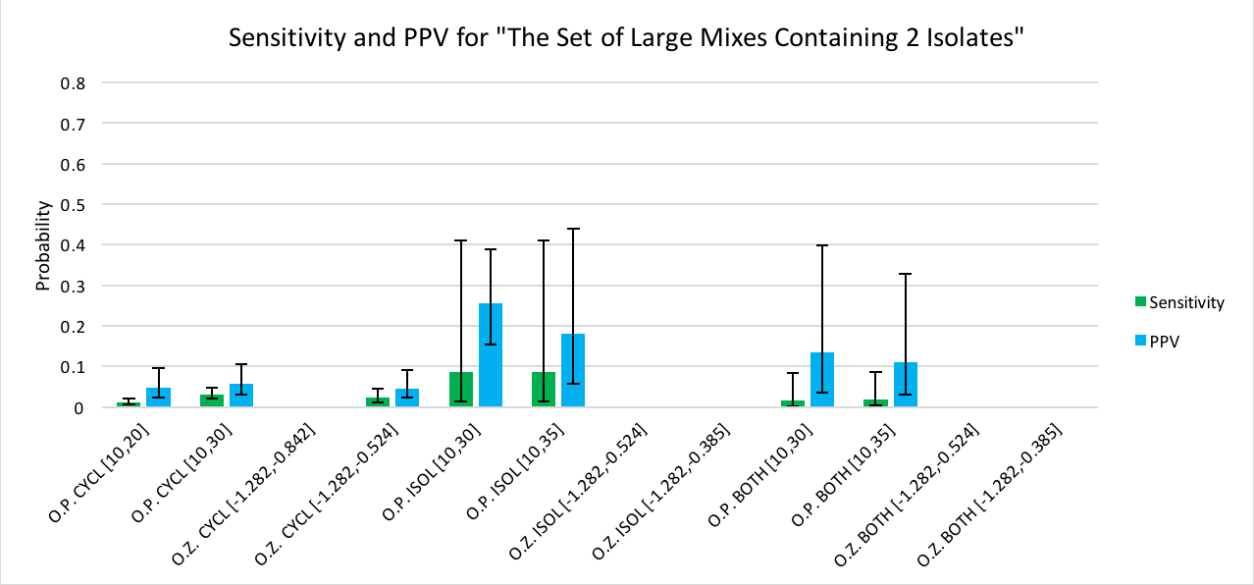

Figure 5.3: The probabilities of the sensitivity and the positive predictive value for the set of large mixes containing 2 isolates. The probability as determined by STATA 15 is represented on the y-axis, and an abbreviated set of each of the SepSIS run conditions for the group of mixtures is represented in the x-axis. The error bars represent the 95% confidence intervals for the results. The run conditions are abbreviated as follows: O.P. = ORGANIC_P RUNMODE, O.Z. = ORGANIC_Z RUNMODE, CYCL = CYCLIC SUBMODE, ISOL = ISOLATED SUBMODE, BOTH = BOTH SUBMODE, and the final conditions are the Min_Score_Value and Max_Score_Value represented as [Min_Score_Value, Max_Score_Value].

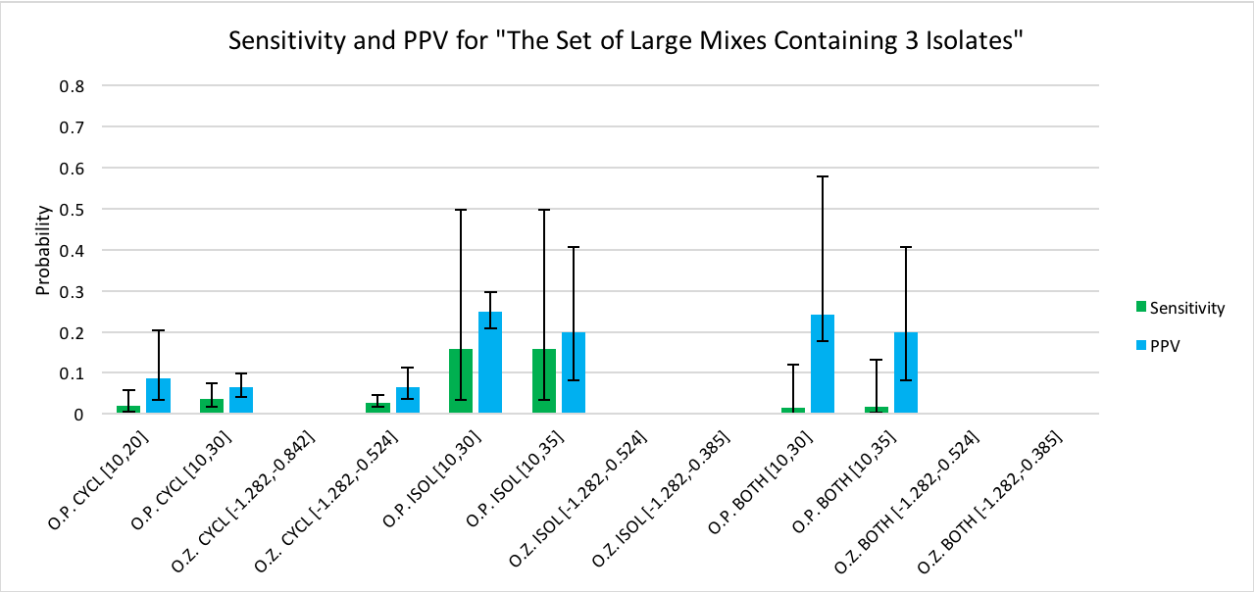

Figure 5.4: The probabilities of the sensitivity and the positive predictive value for the set of large mixes containing 3 isolates. The probability as determined by STATA 15 is represented on the y-axis, and an abbreviated set of each of the SepSIS run conditions for the group of mixtures is represented in the x-axis. The error bars represent the 95% confidence intervals for the results. The run conditions are abbreviated as follows: O.P. = ORGANIC_P RUNMODE, O.Z. = ORGANIC_Z RUNMODE, CYCL = CYCLIC SUBMODE, ISOL = ISOLATED SUBMODE, BOTH = BOTH SUBMODE, and the final conditions are the Min_Score_Value and Max_Score_Value represented as [Min_Score_Value, Max_Score_Value].

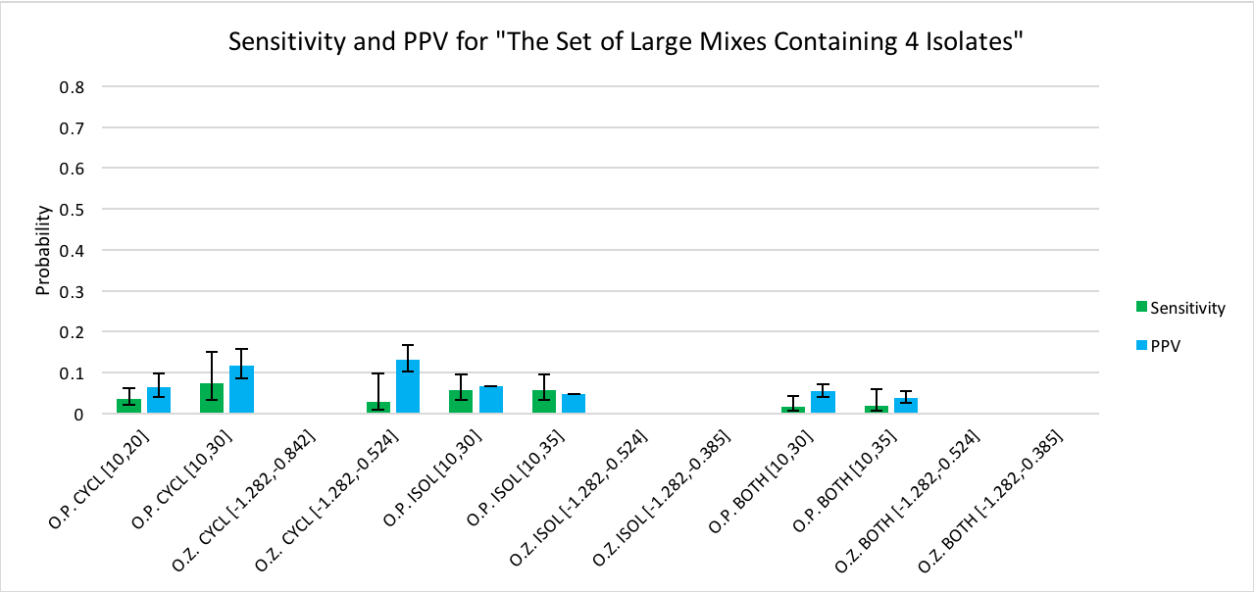

Figure 5.5: The probabilities of the sensitivity and the positive predictive value for the set of large mixes containing 4 isolates. The probability as determined by STATA 15 is represented on the y-axis, and an abbreviated set of each of the SepSIS run conditions for the group of mixtures is represented in the x-axis. The error bars represent the 95% confidence intervals for the results. The run conditions are abbreviated as follows: O.P. = ORGANIC_P RUNMODE, O.Z. = ORGANIC_Z RUNMODE, CYCL = CYCLIC SUBMODE, ISOL = ISOLATED SUBMODE, BOTH = BOTH SUBMODE, and the final conditions are the Min_Score_Value and Max_Score_Value represented as [Min_Score_Value, Max_Score_Value].

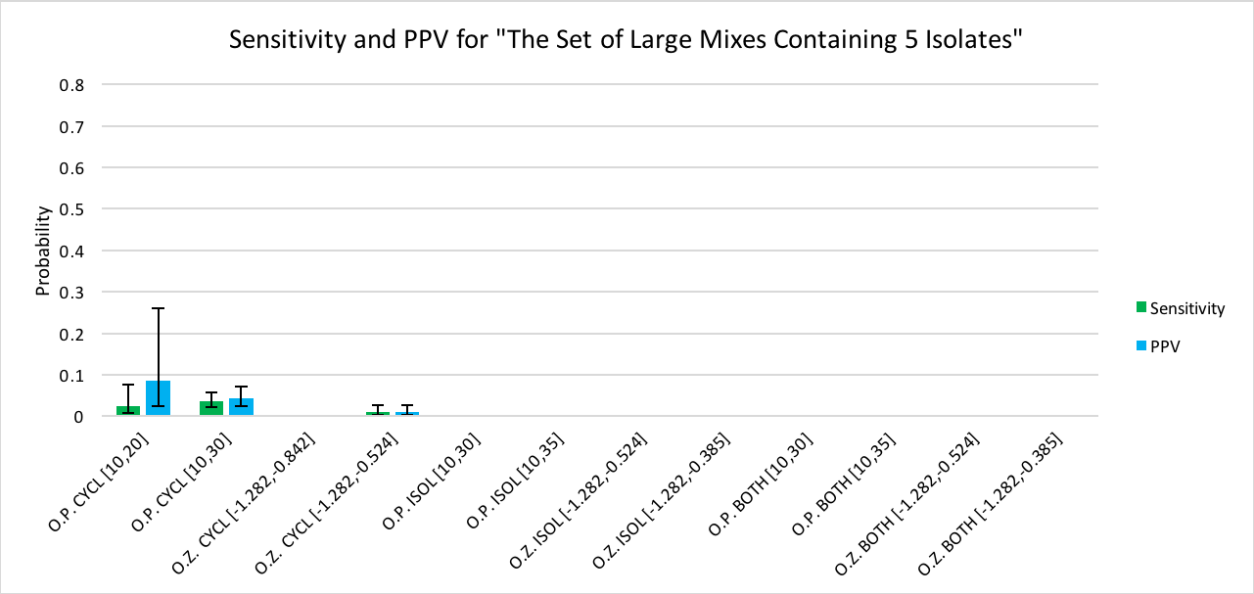

Figure 5.6: The probabilities of the sensitivity and the positive predictive value for the set of large mixes containing 5 isolates. The probability as determined by STATA 15 is represented on the y-axis, and an abbreviated set of each of the SepSIS run conditions for the group of mixtures is represented in the x-axis. The error bars represent the 95% confidence intervals for the results. The run conditions are abbreviated as follows: O.P. = ORGANIC_P RUNMODE, O.Z. = ORGANIC_Z RUNMODE, CYCL = CYCLIC SUBMODE, ISOL = ISOLATED SUBMODE, BOTH = BOTH SUBMODE, and the final conditions are the Min_Score_Value and Max_Score_Value represented as [Min_Score_Value, Max_Score_Value].

5.1.4 The Sets of Large Mixes Containing 2, 3, 4, or 5 Isolates

The results from the runs on the sets of large mixes containing 2, 3, 4, or 5 isolates are presented in Figures 5.3, 5.4, 5.5, and 5.6 respectively. The accompanying data for those runs is presented in Tables A.6 and A.13, A.7 and A.14, A.8 and A.15, and A.9 and A.16.

Figures 5.3 and 5.4 show similar results across all run types. The ORGANIC_Z runs generally failed to produce any results across all runs, except for the ORGANIC_Z CYCLIC [-1.282,-0.524]. This trend continued into the mixes of 4 and 5 isolates as well. The ORGANIC_Z CYCLIC [-1.282,-0.524] runs performed poorly across all runs with sensitivities and PPVs below 0.10 in all cases but the set of large mixes containing 4 isolates, where the PPV was 0.13. The mixes of 2 and 3 isolates had similar trends across the ORGANIC_P ISOLATED and BOTH runs. Sensitivity remains constant with Max_Score_Value changes, meaning that increasing the coverage range did not increase the number of identified sequences for those runs. The PPV did go down on the higher Max_Score_Value runs, also indicating that pool of sequences identified increased with sequences wrongly identified as strain-specific. The 95% confidence intervals had large ranges for these results, decreasing the overall reliability of their values. Lastly, the ORGANIC_P CYCLIC runs produced overall poor results, with low PPVs and sensitivities across all the runs.

In Figure 5.5, there were sharp drops in the PPVs and sensitivities from the ORGANIC_P ISOLATED and BOTH runs, down to below 0.10 on all values. There were no large changes in the PPVs and sensitivities of the ORGANIC_P CYCLIC runs. The ORGANIC_P CYCLIC [10,30] run had an increase in the PPV and sensitivities by 0.05, but the overall values were still quite low. Lastly, Figure 5.6 shows no positive results for the ISOLATED or BOTH runs. The ORGANIC_P CYCLIC and ORGANIC_Z CYCLIC [-1.282,-0.524] runs all produced low numbers of positive results.

Note that the mixes containing larger numbers of isolates had fewer total mixes. The were 15 mixes of 2 isolates, 9 mixes of 3 isolates, 6 mixes of 4 isolates, and only 3 mixes of 5 isolates. This smaller set size may be a partial cause of the poorer results in higher mixes. However, it was more likely that the larger mix sizes skew coverage heavily in such a way that shunts the relative coverage levels of true strain-specific sequences down below the Min_Score_Value cutoffs. Unfortunately, lowering the Min_Score_Value cutoff any more also floods the results pool with many more poorly assembled sequences, and low coverage strain-independent sequences.

5.2 Results of the Evaluation of the Ability of SepSIS to Discern True Strain-Specific Sequences in Larger Mixes

The purpose of this experiment, described in Section 4.5.5 and performed in order to fulfill the goal of Section 3.3.2, was to discern the proportion of strain-specific sequences identified by the verification method that were falsely described as strain-specific. This was performed by attempting to identify and quantify the

Table 5.1: The number of sequences produced by SepSIS that were shared and not shared among a single isolate's different mixes across mix sizes.

Number of Isolates in the Mix	Number of Mixes Total	Percentage of Unshared Sequences	Number of Sequences Shared Among the Isolate	Number of Unshared and Potentially Erroneous Sequences
3	9 Mixes	40.07%	1292	864
4	6 Mixes	48.43%	592	556
5	3 Mixes	31.42%	155	71

number of sequences identified as strain-specific for each strain that do not appear in other multi-strain mixes. Theoretically, a sequence identified as strain-specific from a strain that appears in a mix of 3 strains should also appear in a mix of 2 strains. However, this was not the case in the results. The percentage of unshared strain-specific sequences, the number of strain-specific sequences that are shared between different mixes, and the number of strain-specific sequences not shared between mixes for a strain are reported for all mixes in Table 5.1. As can be seen in the table, a large percentage of strain-specific sequences for each size of mixes could be erroneous. The rate of possible erroneous strain-specific sequences increases from mixes of 3 strains to mixes of 4 strains. In mixes of 5 strains the percentage of erroneous sequences drops, likely due to both the smaller sample size, and low number of sequences identified. Therefore, it can be concluded that SepSIS does not have strong reliability to produce true strain-specific sequences in larger mixes.

5.3 Results of the Investigation into the Possible Existence of Multiple Strains of *M. bovis* on a Single Culture Plate

As described in Section 4.5.6 and in fulfillment of the goal in Section 3.4.1, the possible existence of multiple genetically unique strains on a single culture plate was investigated. The results from evaluating the existence of multiple isolates of *M. bovis* on a single culture plate is broken down into the 3 cases, described in Section 4.5.6 and presented in Table 5.2 and Table 5.3. The results are presented in two tables for space, and have been divided based on isolate names. In those tables, the isolates with MPLM IDs have the following names designating their plate of origin: MPLM_5, MPLM_6, MPLM_45, MPLM_46, MPLM_90, MPLM_91. The individual isolates were designated with an identifier from 1 – 5 after the plate name, such as MPLM_5.1, MPLM_5.2, etc. The results for the MPLM isolates are presented in Table 5.2. The MJ isolates have IDs MJ121 through 140. Isolates MJ121 – MJ125 were picked from a single plate, MJ126 – MJ130 were picked from another, and the pattern continues with MJ131 – MJ135 and MJ136 – MJ140. The results from the MJ isolates are presented in Table 5.3. Isolates that fall under case 1 with no common strain-specific sequences between mixes are not presented in the Table 5.2 and Table 5.3 for space. Table 5.2 and Table 5.3 present

2 values in addition to the case type: the number of strain-specific sequences common across multiple mix sizes, and the total number of strain-specific sequences identified in that mix size for that isolate.

Of the 50 isolates evaluated, 19 were determined to be case 1-s, 15 were case 2-s, and 16 were case 3-s. To elaborate, 19 of the isolates had no or few strain-specific sequences and were either identical to another isolate in the mix, or the sequences representing the isolate in question were a proper subset of the sequences from another isolate on the plate. The 15 isolates that were classified as case 2 have some strain-specific sequences that were unique to that isolate, but also shared strain-specific sequences with at least one other isolate from the same plate. The 16 case 3 isolates possessed strain-specific sequences that were unique relative to all other isolates from the same plate. Independently examining the different plates showed that all but the plate containing MJ126 - MJ130 had at least one instance of a case 3 strain. Therefore, it can be concluded that isolates existing on a single culture plate were non-clonal.

5.4 Results of the Evaluation of the Effect of Contamination on the SepSIS Pipeline

The goal described in Section 3.4.2 was to evaluate the effects of contamination on the SepSIS pipeline. The methodology is described in Section 4.5.7. To determine the effect of contamination on the SepSIS pipeline customized for *M. bovis*, 12 mixes of strains were prepared. These mixes consisted of 4 strains of *M. bovis* mixed pairwise with strains of *M. bovirhinis*, *M. arginini*, and *M. agalactiae*. Sequences belonging to non-*M. bovis* species were identified by separating strain-specific sequences based on the strain IDs. Those strain-specific sequences were then BLASTN-ed against reference genomes for the species in the mix. The *M. arginini* and *M. bovirhinis* strain mixes produced no strain-specific sequences for the non-*M. bovis* species, meaning that no contaminant sequences for those species remained after the post-processing steps. Mixes including the *M. agalactiae* strain did have $5 - 10$ sequences remaining per mix. The identifiers at the beginning of the sequence headers belonged to the *M. agalactiae* strain. However, none of these sequences mapped to *M. agalactiae*. Instead, they mapped to the *M. bovis* reference genome. Therefore, it could be that these sequences are unique to that particular strain of *M. agalactiae* and that the post-processing steps effectively removed contaminate sequences.

During the post-processing steps, one step is focused primarily on identifying possible *S. maltophilia* contamination. Of all the mixes analyzed by SepSIS, only 3 mixes were identified that possessed sequences produced as possible contamination with *S. maltophilia*. The first mix contained 5 strains from group C in the List of Mixes (Section 4.5.1). Only 1 sequence in the mix was flagged for possible *S. maltophilia* contamination. Given that this contaminating sequence was not detected in any of the strains comprising the mix, it is likely that this sequence was falsely assembled rather than a true contaminant. The second mix is from group B in the List of Mixes. This mix was of 3 random strains and the contaminating sequence appears in no other mixes with this strain. Again, it is likely that this sequence was falsely assembled.

Table 5.2: The number of isolates out of the five picked from a plate that had strain-specific sequences present in multiple mixes. Isolates that had no strain-specific sequences are not represented on the table, but their IDs can be derived from the description of the isolates at the beginning of Section 5.3. The numbers are present to help quantify the number of strain-specific sequences that exist, as well as give perspective on the ratio of sequences that remain specific across different mix sizes.

Isolate Name	Shared Sequences Against the Sum of All Sequences From Mix Size 2	Shared Sequences Against the Sum of All Sequences From Mix Size 3	Shared Sequences Against the Sum of All Sequences From Mix Size 4	Shared Sequences Against the Sum of All Sequences From Mix Size 5	Case Type
MPLM_5.1	43/354 (12%)	24/155 (15%)	10/61 (16%)	0/2 (0%)	Case 2
MPLM_5.2	35/300 (12%)	15/113 (13%)	8/50 (16%)	2/3 (67%)	Case 3
MPLM_5.3	46/463 (10%)	24/305 (8%)	10/125 (8%)	1/7 (14%)	Case 3
MPLM_5.4	19/191 (10%)	4/27 (15%)	5/106 (5%)	0/9 (0%)	Case 2
MPLM_5.5	13/180 (7%)	6/40 (15%)	3/19 (16%)	1/7 (14%)	Case 3
MPLM_6.1	1/56 (2%)	1/46 (2%)	0/26 (0%)	0/2 (0%)	Case 1
MPLM_6.2	32/409 (8%)	12/147 (8%)	5/74 (7%)	0/2 (0%)	Case 2
MPLM_6.3	380/1184 (32%)	221/645 (34%)	89/310 (29%)	6/15 (40%)	Case 3
MPLM_6.5	94/224 (42%)	24/56 (43%)	8/18 (44%)	8/15 (53%)	Case 3
MPLM_45.4	70/169 (41%)	21/39 (54%)	20/44 (45%)	9/18 (50%)	Case 3
MPLM_46.1	5/30 (17%)	6/94 (6%)	1/18 (6%)	0/7 (0%)	Case 2
MPLM_46.2	220/395 (56%)	82/126 (65%)	58/92 (63%)	17/24 (71%)	Case 3
MPLM_90.1	54/658 (8%)	54/218 (25%)	21/102 (21%)	4/33 (12%)	Case 3
MPLM_90.3	8/148 (5%)	6/65 (9%)	3/27 (11%)	1/9 (11%)	Case 2
MPLM_90.4	59/232 (25%)	18/55 (33%)	30/114 (26%)	16/47 (34%)	Case 3
MPLM_90.5	20/85 (24%)	26/80 (33%)	14/33 (42%)	1/28 (4%)	Case 3
MPLM_91.2	291/980 (30%)	111/384 (29%)	39/228 (17%)	4/11 (36%)	Case 3
MPLM_91.3	51/385 (13%)	27/221 (12%)	10/98 (10%)	0/4 (0%)	Case 2
MPLM_91.4	3/83 (4%)	0/4 (0%)	1/23 (4%)	0/1 (0%)	Case 1
MPLM_91.5	6/52 (12%)	8/12 (66%)	0/5 (0%)	0/1 (0%)	Case 2

Table 5.3: A continuation of Table 5.2 showing the number of isolates out of the five picked from a plate that had strain-specific sequences present in multiple mixes. Isolates that had no strain-specific sequences are not represented on the table, but their IDs can be derived from the description of the isolates at the beginning of Section 5.3. The numbers are present to help quantify the number of strain-specific sequences that exist, as well as give perspective on the ratio of sequences that remain specific across different mix sizes.

Isolate Name	Shared Sequences Against the Sum of All Sequences From Mix Size 2	Shared Sequences Against the Sum of All Sequences From Mix Size 3	Shared Sequences Against the Sum of All Sequences From Mix Size 4	Shared Sequences Against the Sum of All Sequences From Mix Size 5	Case Type
MJ121	6/99 (6%)	0/31 (0%)	0/10 (0%)	0/1 (0%)	Case 2
MJ123	38/159 (24%)	4/68 (6%)	6/17 (35%)	0/1 (0%)	Case 2
MJ125	43/108 (40%)	18/50 (36%)	4/14 (29%)	6/15 (40%)	Case 3
MJ126	4/34 (12%)	2/33 (6%)	1/10 (10%)	0/2 (0%)	Case 2
MJ127	39/148 (26%)	13/43 (30%)	8/26 (31%)	0/1 (0%)	Case 2
MJ128	2/48 (4%)	0/28 (0%)	0/12 (0%)	0/2 (0%)	Case 1
MJ129	23/110 (21%)	6/22 (27%)	1/4 (25%)	0/0 (N/A)	Case 2
MJ130	19/62 (31%)	11/32 (34%)	0/6 (0%)	0/2 (0%)	Case 2
MJ131	16/111 (14%)	10/38 (26%)	2/15 (13%)	0/6 (0%)	Case 2
MJ133	1/32 (3%)	1/14 (7%)	0/3 (0%)	0/0 (N/A)	Case 1
MJ135	12/56 (21%)	5/12 (42%)	0/1 (0%)	3/6 (50%)	Case 3
MJ136	269/703 (38%)	244/564 (43%)	66/221 (30%)	24/47 (51%)	Case 3
MJ137	78/688 (11%)	28/217 (13%)	12/85 (14%)	2/23 (9%)	Case 3
MJ138	28/343 (8%)	26/258 (10%)	8/54 (15%)	0/6 (0%)	Case 2
MJ139	102/605 (17%)	26/148 (18%)	23/93 (25%)	8/44 (18%)	Case 3
MJ140	72/550 (13%)	62/232 (27%)	17/71 (24%)	4/16 (25%)	Case 3

Table 5.4: The number of isolates out of 27 that shared a single strain-specific sequence for each phenotype. For example, a total of 10 lung specific sequences are shared by 4 lung isolates of *M. bovis*. Note that the 4 isolates are not the same for each sequence. Further details are available in Appendix A, Tables A.17 and A.18.

Number of Isolates	Number of Lung-Specific Sequences Shared Among the Stated Number of Isolates	Number of Stifle-Specific Sequences Shared Among the Stated Number of Isolates
3 of 27	33	8
4 of 27	10	1
5 of 27	5	0
6 of 27	1	0
7 of 27	1	0
8 of 27	1	0

The third mix that was flagged for contamination was the most likely suspect for *S. maltophilia* contamination. This mix was of *M. bovis* isolates from a lung and joint pair described in group C in the List of Mixes. The mix was named MPLM_37.1_and_MPLM_38.1 and produced 4 sequences isolated as strain-specific that mapped with 100% ANI to *S. maltophilia* and less than 94% ANI to any *M. bovis* genome. Further exploration into quality statistics of an independent de novo assembly of the MPLM_38.1 strain showed abnormally high total contig length. The assembly had a contig length of 10 million for MPLM_38.1, while the average for *M. bovis* strains is approximately 3 million. The assembly also had a high NG50 of 100,000 for MPLM_38.1, and the average for *M. bovis* is approximately 30,000. These abnormal statistics indicate the possibility of sample contamination. The sequences flagged as contaminants were removed from further analysis for that mix.

5.5 Results of the Analysis of Paired Lung and Joint *M. bovis* Isolates for Tropism-Specific Sequences

To show that the SepSIS validation method is applicable as a method for contrasting read sets with differing phenotypes, SepSIS was applied to the task of finding *M. bovis* genotypic sequences specific to a particular tissue tropism. This goal is described in Section 3.4.3, and the methodology used is described in Section 4.5.8. This goal was performed using mix set D, as discussed in Section 4.5.1, containing pairs of isolates cultured and sequenced from lung and stifle joints, with each pair originating from a single animal. By mixing reads from *M. bovis* isolates, SepSIS identified and isolated sequences specific to either isolate in a pair mix. Through further comparisons, sequences specific to particular tropisms were identified.

Table 5.5: The number of tropism-specific sequences that mapped to a particular gene for each tropism using BLASTN. The 33 lung sequences shared between 3 isolates each are not included in this list. Note that the number of sequences listed is independent information from the number of isolates containing each sequence, which is presented in Table 5.4. Each individual sequence mapped to a variant of the genes presented here. Further details on which sequences mapped to which gene are presented in Appendix A, Tables A.17 and A.18.

Matching *M. bovis* Gene	Number of Lung-Specific Sequences Mapping to the Gene	Number of Stifle-Specific Sequences Mapping to the Gene
Variable Surface Lipoproteins	8	4
HAD-superfamily hydrolase	9	4
ISMbov1	3	3
ISMbov2	7	3
ISMbov3	1	1
Putative Lipoprotein Protein	1	1
deoxyribodipyrimidine photolyase (uvrC), transposase	1	0
Only Full Genome Matches	0	1
Total Number of Evaluated Sequences	18	9

Of the 29 isolate-pair mixes, pairs with identifiers MPLM_99.2_and_MPLM_100.5 and MPLM_105.4_and_MPLM_106.5 did not produce any strain-specific sequences. The sequences produced by the remaining 27 pairs were successfully compared to identify sequences shared across the lung-specific and stifle-joint-specific sequences. A total of 52 sequences were shared by 3 – 9 lung isolates and 9 sequences were shared by 3 – 4 joint isolates. Of these lists, a maximum of 8 lung isolates shared a single strain-specific sequence, and a maximum of 4 stifle-joint isolates shared a single strain-specific sequence. The total number of sequences listed is represented in Table 5.4. The strain-specific sequences and associated data are present in Appendix A, Table A.17 and Table A.18. To identify these sequences they were BLASTN-ed against the NCBI nr database, and the most common matches reported [44]. This process did not include the 33 lung sequences shared by 3 isolates. The most common genes matches from the search results are represented in Table 5.5. The 19 lung sequences and 9 stifle joint sequences tested most commonly to have 98% – 100% ANI to the following genes: HAD-superfamily hydrolase, variable surface lipoproteins (VSPs), and ISMbov insertion sequences.

6 Discussion

The discussion chapter contains 4 sections. Section 6.1 contains a brief description of the history of the SepSIS's development and the reasoning behind it. Section 6.2 is a discussion of the coverage-based method of SepSIS, the evaluation results, parameter selection, and how the coverage-based method relates to graph design. Section 6.3 contains a discussion of the verification method, and the results of the independent analyses conducted. Section 6.4 contains a comparison to other similar tools.

6.1 The Creation of the SepSIS Pipeline

6.1.1 Creation of SepSIS

The development of SepSIS in accordance with goals of Section 3.1 started with the idea of attempting to find sequences that have alternate forms, with each form containing different internal subsequences. This concept is discussed at beginning of Section 4.1.1. In theory, a SPAdes assembly graph is an ideal place to look for subsequences specific to a strain in an assembly due to the alternate subsequences that start or end at branch nodes. The idea behind using the SPAdes assembly graph for SepSIS was inspired by an add-on package to the SPAdes assembler called Recycler [5, 41]. Recycler's algorithm functions by parsing the assembly graph produced by SPAdes and extracting small cyclic components within the graph that are likely to represent plasmids. Recycler also takes a BAM file of the short reads mapped against the assembly graph as input. This was used to roughly calculate coverage uniformity between subsequences as a method to determine if adjacent subsequences with similar coverage levels comprise a plasmid. Examples from Recycler and SAVAGE [4] were the particular inspiration to create a coverage-based method.

SepSIS evolved through a large number of small changes and bug fixes, but there were two major iterations of the method for evaluating strain-specificity before the current method of SepSIS was developed. The first iteration was focused on determining if the coverage similarities between adjacent subsequences truly exist. To do this, the coverages for the strain-independent and strain-specific nodes were identified by finding strain-specific subsequences that started or ended at a primary node representing a strain-independent subsequence. This was performed by using the BAM file of reads with IDs mapped against the assembly graph to extract subsequence coverage levels. In theory and assuming uniform coverage for two different isolates of *M. bovis* in a mix, a strain-specific subsequence should have half the coverage of a strain-independent subsequence. This would be due to the strain-independent subsequences being assembled with reads from 2 isolates, instead of one. However, upon testing it was determined that this was not the case due to highly variable coverage

in the *M. bovis* datasets. There was seemingly very little relationship between coverage levels and strain-specificity. This can be somewhat seen in the poor results of Section 5.1. Despite this, an attempt at creating a locally calculated strain-specificity using dynamically generated cutoffs was attempted. Local Z-Scores for subsequences were calculated for a small set of subsequences, and subsequences not meeting these threshold were excluded. This was less successful than the results shown Section 5.1.1 and was abandoned.

Given the success of other similar coverage-based tools, as discussed in Section 6.4, development of SepSIS continued. The second iteration on SepSIS evolved into the verification method of SepSIS to extract strain-specific subsequences using metadata, meeting the requirement for goal 3.1.2, and also report some basic statistics on the coverage for those strain-specific subsequences. It was at this stage in development that the subsequence path isolation functions were developed in order to comprehensively extract all possible variations of strain-specific sequences. The graph parsing functions went through dozens of iterations to reach the current form to improve time complexity, and ensure that all the permutations of sequences meeting the strain-specific criteria were expressed. The improved graph parsing and verification method found that the average and modal coverages of the strain-specific sequences were indeed lower than the coverages of strain-independent subsequences. However, the distribution of coverages seemed to vary in an extreme manner from *in silico* mixture to *in silico* mixture. This investigation gave way to the current coverage-based mode ORGANIC_Z that uses Z-scores. However, with Z-scores there is the assumption of a normal distribution, which is not always the case. Thus the ORGANIC_P percentile mode was created to attempt to compensate for the highly variable distribution. These modes were created to satisfy goals 3.1.1 and 3.1.3. Lastly, the strain-independent subsequences were included in the output of all methods to satisfy the goals discussed in Section 3.1.4.

6.2 Results from the Evaluation of SepSIS and the Testing to Select the Parameter Settings

The results described in Section 5.1 show the evaluation of the coverage-based modes of SepSIS in accordance with the goal in Section 3.3.1. Overall, the coverage-based modes were generally unsuccessful at reliably isolating the strain-specific sequences represented in the verification method output set. What follows is an explanation of the results for the set of *in vitro* mixes and all other sets of mixes, a breakdown of possible reasons why the coverage-based method was unsuccessful, and a description of why the parameter settings for the runs were selected.

6.2.1 The Set of *in vitro* Mixes

Unfortunately, there were no positive results when comparing the strain-specific sequences from the set of *in vitro* mixes to the set of *in silico* mixes in Section 5.1.2. A number of BLASTN cutoffs were attempted when comparing two sets of mixes, starting from 99% ANI and iteratively decreasing to 94%. The 94% ANI was

chosen as the minimum threshold due to being the same-species whole genome minimum ANI as discussed in Section 2.2. Given that no matches were found at the threshold, it can be concluded that the output sequences were entirely different for the set of *in vitro* mixes and the set of *in silico* mixes.

The failure of SepSIS to produce the same strain-specific sequences for the set of *in silico* mixes and the set of *in vitro* mixes is likely linked to differences in the sequencing, mixing, and assembly of their read sets. For example, it is possible that the *in vitro* data lacked strain-specific sequences present in the *in silico* data. One reason for this is that the *in silico* datasets were created from an older set of previously cultured and sequenced isolates, while the *in vitro* dataset was made from frozen and re-cultured isolates, physically mixed in lab, and sequenced with a different method. All of these steps may have led to different environmental factors that led to differences in the nucleotides sequenced. It could be that some strain-specific sequences were underrepresented in the final products of the *in vitro* mixed dataset. The quality of the reads produced during sequencing also varies between sequencing runs, possibly affecting the assembly quality and presence of strain-specific sequences in the assembly.

Another possible reason for lack of identical strain-specific sequences between mixes was differences in the assembly of the reads caused by the *in silico* read mixing. No reads were excluded when the reads sets were combined *in silico*. Therefore, the *in silico* read sets were 2X – 5X the size of the *in vitro* read sets resulting in underrepresentation of strain-specific subsequences and overrepresentation of strain-independent subsequences. Its likely the coverage-based modes work better on the *in silico* datasets due to their higher overall coverage, and the skewed coverage distributions created by mixing multiple read sets. For example, if a gene has high coverage in 2 isolates being mixed and is strain-independent, it's coverage will be skewed higher. In the same mixture, a strain-specific sequence has low coverage and it will remain low when mixed. Because the SepSIS parameters were calibrated based on cases such as this, it is logical that if the *in vitro* mixes do not have the same coverage skewing they will not produce similar results.

Random read selection with different ratios of strain was briefly attempted to more accurately model the *in vitro* mixed read sets. For example, if 2 isolates were being mixed, half of the reads from isolate 1 were randomly selected and half of the reads from isolate 2 were randomly selected. Different ratios including 1 third of reads from isolate 1 and 2 thirds of reads from isolate 2 were also attempted. However, these mixes resulted in highly fractured and fragmented assemblies and did a poorer job of modelling the *in vitro* mixed assemblies than combining all reads to create the *in silico* mixes. Therefore, the mixes of isolates created from combining all reads from an isolate were used in this thesis.

6.2.2 The Sets of All Other Mixes

The results of the set of *in silico* mixes, the set of paired isolates mixes and the sets of large mixes containing 2, 3, 4, or 5 isolates are discussed. It is concluded that the overall effectiveness of the coverage-based modes of SepSIS at correctly determining strain-specific sequences is low, but not completely ineffective. The number of possible subsequence combinations from an assembly graph is massive and it is significant

that the mode returned positive results with PPVs and sensitivities in the 0.10 to 0.30 range. However, these low probabilities indicate that the current coverage-based mode is not a practical way of assigning strain-specificity for such a dataset. Additionally, in the results for all sets of mixes, it is notable that the Z-Score-based ORGANIC_Z mode was nearly completely ineffective at discerning strain-specific sequences compared to the percentile-based ORGANIC_P RUNMODE. However, the development of the ORGANIC_Z RUNMODE was fundamental to the SepSIS, therefore the results for that mode are included in this thesis. A brief breakdown of the results from each individual set of mixes follows.

The set of *in silico* mixes are arguably the most successful set overall with the highest PPVs in the CYCLIC runs on the CSCCs of any set, and strong results for the BOTH runs on the whole graph. There is a likely reason for this. The isolates comprising the set of *in silico* mixes are isolates from an older dataset, developed previously to this thesis, and these isolates were specifically selected for the purpose of comparison to new sequenced, *in vitro* mixed isolates. These isolates all were assessed to have higher than 30X read coverage in that dataset, producing good quality independent assemblies. This translates to a higher quality assembly and better results.

The set of paired isolates mixes was meant to assess SepSIS's ability to process highly similar isolates. The results from the set of paired isolates mixes were worse than the set of *in silico* mixes, but were very similar to the results for the set of large mixes containing 2 isolates. The only difference between the two sets is that the ISOLATED and BOTH runs for the set of paired isolates mixes have poorer results than the set of large mixes containing 2 isolates. One possible explanation for this is that the highly similar isolates had fewer high quality ISCCs in the assembly graph. This could have been caused by highly similar isolates having overlapping sequences, resulting in fewer fragmented sections in the assembly graph. Note that some ISCCs had high coverage (30X or greater), but this was rare. ISCCs averaged 2X – 4X coverage per mix, while CSCCs had a much larger range of possible coverages. The relatively similar results between the set of paired isolates mixes and the set of large mixes containing 2 isolates indicated that the results reflect the number of isolates in the mix and the growth and sequencing conditions shared by the isolates used in the mix, rather than the presence of highly similar isolates. Thus, the conclusion is that running SepSIS on mixed, highly-similar isolates produces similar results to mixes of random isolates for CSCCs, but slightly poorer results than randomly mixed isolates for ISCCs and the graph as a whole.

The purpose of running SepSIS on the sets of large mixes containing 2, 3, 4, or 5 isolates was to contrast the results and examine the differences caused by larger numbers of isolates in a mix. Mixes of 2 isolates and 3 isolates had highly similar results. The mixes of 3 isolates showed slightly higher PPVs and sensitivities, up to an increase of 0.10 in the case of PPV for the ORGANIC_P BOTH runs in the mixes of 3 isolates. However, the 95% confidence intervals were large for the ISOLATED and BOTH runs, which detracts from the significance of the results. The conclusion at this stage is that there is very little difference in the coverage-based modes' ability to find strain-specific sequences in 3 isolates when compared to 2.

There was a noteworthy decrease in the sensitivities and PPVs when increasing the number of isolates in

the mix above 3. When looking at the results, a downward shift in all sensitivities and PPVs can be seen. One reason for this is the change in distribution of sequence coverage due to the combination of 4 or 5 isolates. Because the coverage-based modes are partially based on the coverage distribution, some of the strain-specific sequences that do exist have coverages below the Min_Score_Value and are not identified. Another reason for the downward shift in sensitivities and PPVs is simply a lack of strain-specific sequences when a large number of isolates are mixed. The total number of true strain-specific sequences drops precipitously at 5 isolates in a mix. This is shown in Section 5.2.

It is concluded that the ORGANIC_P mode generally produces results with higher and more reliable PPVs and sensitivities than the ORGANIC_Z, except on runs with the SUBMODE set to ISOLATED. The ISOLATED and BOTH SUBMODEs seemed to have the highest variability between runs and datasets. The CYCLIC mode seemed to produce the most consistent results, but the PPVs and sensitivities were never high enough to justify use of the tool on true datasets for study, especially considering the failure of *in silico* mixes to model the *in vitro* mixes. The effects of changing the Max_Score_Value were inconsistent between runs and modes. In the *in silico* mixes the higher Max_Score_Value generally had improved scores, but this was not true for all runs. These scores will be discussed in-depth in the next subsection.

6.2.3 Value Selection for the maxPathNodeLength

The maxPathNodeLength value is an internal variable for SepSIS discussed in Section 4.3.7. The variable is an upper limit to the number of adjacent nodes (subsequences) that might be merged during the process of merging nodes into strain-specific sequences. For all results sets analyzed in this this thesis, the variable was set to 8. This setting was determined by examining the results of preliminary experimental runs while developing SepSIS. Runs were tested with maxPathNodeLength initially set to 20 and iteratively reduced to 6 by a step size of 1. When set to values of $12 - 20$, individual runs of SepSIS took tens of minutes to hours to complete. This was too long to be practical for hundreds of runs. Values of $9 - 11$ took tens of minutes to run, while values of $6 - 8$ took less than 10 minutes to run. It was noted at a value of 6 that some strain-specific sequences became truncated, while the results sets from $7 - 11$ were identical. Therefore, the value of 8 was chosen to prevent output sequences from being truncated and to prevent long runtimes.

6.2.4 Value Selection for the Min_Score_Value and Max_Score_Values

Selection of the specific Min_Score_Value-s and Max_Score_Value-s required experimentation. These experimental runs to calibrate these values were an attempt at customizing SepSIS for variable coverage, as discussed in Section 3.1.3. When evaluating mixtures of isolates, each mixture has its own optimal Min_Score_Value-s and Max_Score_Value-s. However, when performing a batch run, it is much more practical to set a single set of parameter values for the batch. Note that because the ORGANIC_Z RUNMODE was much more ineffective than the ORGANIC_P RUNMODE, the ORGANIC_P percentile parameters for Min_Score_Value and Max_Score_Value will be the primary focal point of discussion in this subsection.

The current form of the graph parsing functions of SepSIS originally only used a Max_Score_Value threshold and no Min_Score_Value. This was corrected because a lack of a lower boundary cutoff drastically increased in the number of subsequences falsely identified as strain-specific during testing. The Min_Score_Value was iteratively increased from 0 by 2.5 percentile and the roughly equivalent Z-Score value until it reached the 10th percentile and -1.282 Z-score. Above those scores was the point at which true positives started to be heavily removed from the test set. Therefore, the Min_Score_Values were set at the current threshold.

The Max_Score_Values for the CYCLIC runs were set to the 20th and 30th percentiles primarily for a contrast between the two. Values much higher than the 30th percentile started giving much larger numbers of false positives, as can be seen in the shift from the 20th percentile to the 30th percentile. This heightened the sensitivity of the run, but lowered the PPV. The ISOLATED and BOTH runs had higher Max_Score_Values due to the much larger number of low coverage subsequences present in the ISCCs and whole assembly graph. The ISCCs primarily consist of poorly assembled sequences with low coverage, and therefore they skewed the coverage distribution. The higher Max_Score_Values were an attempt to counteract this phenomenon.

The runs with the BOTH SUBMODE seem to have a hard cap at approximately the 40th percentile and a Z-Score of -0.2019 that is roughly equivalent to the 42nd percentile. At this point, the pathing algorithm within SepSIS experiences a drastic increase in the number of paths predicted, and runtime and memory usage increase in turn. While runtime is not an evaluated statistic in this thesis, generally SepSIS takes $1 - 5$ minutes to run on a mixture on any mode. However, SepSIS relies on the strain-specific criteria to extract a subset of paths through the assembly graph that represent the sequence. If the user increases the criteria to be too broad, the algorithm attempts to extract an exponentially higher number of paths, drastically increasing runtime and memory usage. It is for these reasons that the given Min_Score_Value-s and Max_Score_Value-s were chosen.

6.2.5 Conclusions for the Graph-Based Design

The CYCLIC runs on CSCCs, ISOLATED runs on ISCCs, and BOTH runs on the whole assembly graph all have differing probabilities for PPVs and sensitivities, but there are some trends among and across mix sets. The ISOLATED runs give the most inconsistent runs across different mixes, with widely varying sensitivities and PPVs, as well as large 95% confidence intervals. The runs of BOTH and CYCLIC often have similar sensitivities and PPVs, which are low on all runs. Generally, there is not a large difference between the CSCC results and the BOTH results because there are relatively few strain-specific sequences in the ISCCs when compared to the CSCCs.

The graph-based design of SepSIS that relies upon the assembly graph is both a great asset and a massive obstacle. A CSCC within the graph could, in theory, represent a perfected assembled cyclic bacterial chromosome. In practice, this is not the case due to the complexities of genome assembly in SPAdes. As discussed earlier, in a theoretical situation, a strain-specific subsequence in a mix of strains would have a coverage proportional to the number of strains in the mix. This theory can be applied to SNP and SNV

ratios, and is in tools discussed in Section 6.4. However, in practice, the theory does not apply well to the assembly graph produced by SPAdes.

6.3 Analysis of Verification Method Output

After experimenting with the coverage-based analysis, it became clear that further analysis of the *M. bovis* dataset would need to proceed with the verification method of SepSIS to produce reliable results. However, there is an innate problem with the verification method. As mentioned in Section 4.5.3, SPAdes may create duplicate sequences during assembly. The verification method of the SepSIS pipeline relies on the mapping of reads with identifiers by minimap2 against the assembly graph produced by SPAdes. Minimap2 has the capacity to map a single read to multiple regions of an assembly, but it does not reliably do so. This can result in reads from only one strain mapping to a subsequence when there exists reads from multiple strains that should map to that subsequence. This subsequence would then be falsely evaluated as strain-specific by SepSIS. If two duplicate sequences are separately falsely evaluated as strain-specific by SepSIS, then the duplicate appears in the SepSIS output. This is the reason that a post-processing step was implemented to remove duplicate sequences.

6.3.1 The Evaluation of the Ability of SepSIS to Discern True Strain-Specific Sequences in Larger Mixes

The flaws of SPAdes and minimap2 prompted further investigation into how well the verification method of SepSIS functioned with larger mixes of isolates. This is the goal described in Section 3.3.2. This was performed by investigating the number of sequences specific to a single isolate (strain) in the mix, in mixes containing 3, 4, and 5 isolates. If a strain-specific sequence appeared in any of the other mixes, it was flagged as a shared sequence. If the strain-specific sequence did not appear in any other mixes, it was flagged as potentially erroneous. The mixes of 2 are not presented because the number sequences not shared by any other mixes would not be relevant. This is because a sequence that is strain-specific to 1 strain in a mix of 2 will not be flagged as strain-specific in a mix containing 3 or more strains due to the presence of that sequence within another strain in the mixes.

In the results described in Section 5.2, it can be seen that a large percentage of the sequences do not appear in other mixes, and could potentially be erroneously assembled or falsely flagged as strain-specific. This inaccuracy could be due to a number of reasons. As more isolates are added to a mix, the ability of SPAdes to create a coherent assembly will decrease. Additionally, the presence a large number of highly-similar sequences in the output assembly graph may create difficulties for minimap2 to map reads to the assembly graph subsequences. SepSIS relies on this mapping to identify sequences as strain-specific. Therefore, if minimap2 fails, SepSIS will fail. It can be concluded that the verification method of SepSIS and the preprocessing steps do not handle larger mixes well.

6.3.2 The Existence of Multiple Strains of *M. bovis* on a Single Culture Plate

The first experimental objective of this thesis from Section 3.4.1 was to investigate whether multiple strains of *M. bovis* exist on a single culture plate at one time. It is known that unless an isolated single cell is allowed to grow, there is a chance of differing strains of a single species creating non-clonal culture. One notable paper describes how a contaminant strain of *Geobacter sulfurreducens* persisted in a culture primarily consisting of 1 other strain across multiple cultures and studies of that isolate [45].

The investigation using SepSIS, presented in Section 5.3, came to a similar conclusion, that indeed there is evidence that multiple strains of *M. bovis* do exist on a single culture plate and that a single isolate has the potential to contain more than one strain. The experiment presented assumed that the individually sequenced isolates contain only 1 strain. These isolates were mixed *in silico* with the other isolates from the same plate producing results showing the presence of strain-specific sequences. In these mixes, 16 of 50 isolates were evaluated as containing strain-specific sequences when compared to all other isolates on the same plate and an additional 15 of 50 isolates containing sequences specific to a subset of isolates on the plate. Therefore, it is concluded that there is evidence that multiple isolates of *M. bovis* on a single culture plate and this bears further investigation with other genotypic analysis methods.

6.3.3 Implementation of Anti-Contamination Post-processing and Contamination Results

The effects of contamination of the *M. bovis* dataset was a concern from the start of development. Therefore, investigating contamination is a goal from Section 3.4.2. The results discussed were presented in Section 5.4. The first post-processing steps of SepSIS were implemented as a part of an effort to ensure that SepSIS was outputting sequences that were non-contaminated and correctly assembled. This was performed by BLASTN-ing the strain-independent subsequences on the ends of the strain-specific sequences against the existing reference genomes in *M. bovis*. As described in Section 5.4, the post-processing steps included a specific stage to search for possible *S. maltophilia* sequences that mapped to *S. maltophilia* and not to *M. bovis*. There was only one isolate that had any likelihood of *S. maltophilia* contamination: MPLM_38.1. Further exploration into the quality statistics of an independent assembly of that strain showed some anomalies as well. Therefore, it was possible that this one isolate was indeed contaminated with *S. maltophilia*, but the post-processing step was able to identify and remove the contaminating sequences.

The further experimentation on the purposefully contaminated mixes showed that the post-processing steps removed the contaminated sequences. There were no *M. arginini* or *M. bovirhinis* strain-specific sequences and very few *M. agalactiae* sequences as output to the post-processing steps. When examining those sequences, they were above the BLASTN cutoff for the *M. bovis* reference genome, but not above cutoff for the *M. agalactiae* reference genome. This indicates that sequences might have been a strain-independent sequence and were not mapped as such, or the sequence may have appeared in one of the *M. bovis* reference

genomes, but not in the *M. agalactiae* reference genome.

6.3.4 Sequences Associated with the Tissue Tropisms of the Lung and Joint *M. bovis* Isolates

The analysis of phenotype-specific sequences from the paired lung and joint isolates identified 3 primary genes as candidates for affecting tropisms, as according to the goal in Section 3.4.3. The genes are HAD-superfamily hydrolase, variable surface lipoproteins (VSPs), and ISMbov insertion sequences. These results were presented in Section 5.5.

HAD-family hydrolases are a superfamily of enzymes and have a broad range of functions. A recent study conducted on *M. bovis* genomes used specialized software to predict and detect adhesion-related factors and putative adhesins [17]. One of the targets identified as having a high probability of being involved in adhesion was a putative phosphatase (Genbank ID: SBO46364.1), which is a HAD-family hydrolase [17]. It is possible that mutations to HAD-family hydrolases may affect the adhesion of *M. bovis* to particular tissue types.

Somewhat similarly, hydrolysing activity has been discussed in relation to virulence in *Mycoplasma mycoides* subsp. *mycoides*. In a previous study, strains of *M. mycoides* expressing a particular isoform of glucosidase were shown to have lower hydrolysis activity [48]. The authors linked this lower activity to an increased survival rate of *M. mycoides* in environments with high levels of Beta-D-glucosides. Despite this, these strains were also shown to have lower virulence when compared to African strains without the isoform mutation to glucosidase. A mutation to a HAD-family hydrolase sequence is likely capable of affecting virulence and tissue tropism. However, a much more in-depth study into the activity of hydrolase activity in *M. bovis* would be needed to confirm or refute this.

VSPs have been linked to the adhesion of *M. bovis* to host cells, as well as antigenic variation [42, 43]. VSPs contain highly repetitious sequences, which should be noted as a possible source of error during genome assembly that could lead to false identification as strain- or tropism-specific. However, this variation may also affect the ability of *M. bovis* cells to adhere to specific *Bos taurus* cell types, leading to specialized tissue tropisms. Therefore, the subsequences mapping to HAD-family hydrolase and VSP sequences are the most likely targets for further study.

ISMbov insertion sequences have not been linked to possible variations in tissue tropism. One study of 1421 samples from milk, udder, lungs, and nasal swabs showed that ISMbov1 and ISMbov2 types varied greatly between herds of cattle, but did not between infection locations [1]. However, insertion sequences are known to affect virulence and metabolism in cells [47]. Therefore, it is possible that a novel link may exist in the local *M. bovis* dataset.

From these results, there is evidence that these genes in *M. bovis* may influence tissue tropism. Further research should be performed both on the paired *M. bovis* dataset and these genes in particular to establish a stronger link between the genotype and phenotype of the isolates.

6.4 SepSIS in Relation to Other Tools

There are several existing tools similar to SPAdes and the SepSIS pipeline that seek to identify differing strain-specific sequences or haplotypes between strains. ALLPATHS is a *de novo* assembler that is superficially similar to SPAdes that can represent strain-specific sequences [8]. ALLPATHS requires a set of short reads as input and produces a contig output file in the .EFASTA format. The difference between the .EFASTA file and the SPAdes .FASTA contig file is that ALLPATHS does not completely compress the paths through the assembly graph into a single sequence. Instead, subsequence options are presented as subsequences within square brackets and separated by commas within a sequence. For example, `AA[TT,CC]GG` represents both `AATTGG` and `AACCGG`. This is similar to an assembly graph in the ability to represent subsequences with matching start and end points. The major difference between the .EFASTA format and the .FASTG graph is that the .EFASTA does not contain information describing the relationships between highly complex or tangled variants of subsequences. Instead, all subsequences are represented as a linear string of characters. As explained above in Section 6.2.4, this complexity is both an advantage to search for highly unique strain-specific sequences, and a massive complication.

SAVAGE (Strain Aware Viral Genome Assembly) assembles reads into contigs while preserving subsequences and SNPs unique to particular haplotypes that may occur due to presence of multiple viral quasi-species [4]. This is performed by iteratively constructing an overlap graph from overlapping reads in a manner that is functionally similar to the de Brujin and assembly graphs used by SPAdes. During SAVAGE's assembly iterations, co-occurring mutations are identified in overlapping subsequences. They are allowed to remain if they meet stringent frequency and quality cutoffs. A requirement for proper use of SAVAGE is a deep-coverage dataset as input. The paper describing SAVAGE reported using 250-bp paired Illumina MiSeq reads at a coverage depth of 20,000X. This coverage depth allows for a very precise iterative expansion of their overlap graph allowing for these haplotypes to be extracted.

The self-proclaimed first bacterial haplotype assembly method is named the Evolutionary Reconstruction of Haplotypes (EVORhA) [33]. EVORhA is a reference-based method that aligns sequences against a reference genome and finds local haplotype differences. The subsequences containing local haplotypes are extended and merged based on Gaussian distributions that describe the frequency at which individual haplotypes occur. EVORhA requires 150X to 200X sequencing coverage, as reported in their real world samples used for experimentation. During their evaluation test, EVORhA used a simulated 50X coverage input set and only produced a result of 70% haplotype reconstruction accuracy.

The assembler metaSPAdes is an alteration of the SPAdes algorithm that was created for use on metagenomic data [28]. MetaSPAdes was built for the purpose of assembling metagenomic samples that have highly conserved genomic regions and nonuniform coverage levels to isolate related strains with varying abundances. MetaSPAdes functions in a manner similar to SPAdes (discussed in Section 2.10.1) with a few changes. These changes cause the algorithm to isolate paths in the assembly graph that represent unique long genomic frag-

ments within a metagenome. This is performed by adding new conditions when forming the structure of the assembly graph using relative coverage levels. This is significant because these long genomic fragments likely represent species-specific or strain-specific sequences.

SepSIS is similar in purpose to all of the above algorithms, but it differs in some key details. The largest difference is the stage of assembly at which the data is assessed for differing subsequences or haplotypes. The SPAdes assembly graph is assessed by SepSIS after SPAdes has finished assembling the sequences, unlike most of the above algorithms which isolate unique haplotypes during assembly. The metaSPAdes assembler is similar to SepSIS in that it parses the SPAdes assembly graph to produce sequences unique to particular strains. The differences between the two algorithms are stem from their execution and purpose. The metaSPAdes assembler incorporates the strain-specific sequences into variants of contigs for output. SepSIS focuses on identifying, ID tagging, and extracting the strain-specific sequences to allow for the specific study of contrasting sequences.

SepSIS and the above algorithms are similar in their use coverage levels or frequencies to assign strain specificity. However, SepSIS has a reliance on post-assembly reported coverage. This was an implementation design choice that was intended to help expand the scope of finding strain-specific sequences, but ultimately hindered the accuracy of the SepSIS algorithm. The intent with SepSIS was to find larger-scale sequence differences in sequences as well as localized haplotype mutations by searching the entire resulting assembly graph for regions of lower, but still reasonable coverage. In an ideal situation unique strains would be represented by similar coverage levels among subsequences common to a particular strain. However, the scope of the algorithm paired with the coverage-based approach proved to be a major downfall of SepSIS.

7 Future Work and Conclusions

This thesis focused primarily on the development of the SepSIS algorithm using the *M. bovis* dataset. Despite the generally unsuccessful nature of SepSIS, the experiments analyzing the SepSIS verification algorithm results did provide some positive results. There were multiple cases of strain-specific sequences unique to a single strain on a culture plate when comparing colonies on the plate. In a separate analysis, there were multiple sequences unique to the lung tropism or to the stifle joint tropism of *M. bovis*. The possibility of SNPs, genes, or strains unique to *M. bovis* infecting these tissue types is a possibility. This section describes possible further research into these elements of *M. bovis*, as well as the overall conclusions of this thesis.

7.1 Future Work

7.1.1 Further Investigation Into the Existence of Multiple Strains of *M. bovis* on a Single Culture Plate

The investigation of SepSIS into the existence of multiple strains on a single culture plate provided evidence that strains on a single culture plate are non-clonal. This hypothesis can be explored further without the use of SepSIS. By performing a multiple sequence alignment on reference assemblies of the 5 isolates from a single plate, the assembles can be checked for SNP level differences. It can also be checked for larger sequence difference, assuming minimal bias from the reference assembly. In this way it can be proven or disproven that multiple strains exist on a single culture plate.

7.1.2 Further Investigation Into the Sequences Associated with Paired Lung and Joint *M. bovis* Isolates

Investigation into the similarities and differences between strains of *M. bovis* infecting the lung and stifle joint tissues has already taken place to some degree outside of this thesis. In the paper "Comparison of Two Multilocus Sequence Typing Schemes for Mycoplasma bovis and Revision of the PubMLST Reference Method" authored by Dr. Register, there was no clustering of *M. bovis* isolates by tropism when using an MLST analysis [36]. Our own investigation using a multiple-alignment of reference assembled lung and joint isolates and subsequent generation of a maximum-likelihood tree showed no grouping of isolates by tissue tropism. Further investigation should therefore focus on smaller scale differences in the genome. A genome-wide association study (GWAS) searching for SNPs that associate with the particular phenotype

could highlight notable similarities. Additionally, searching for and/or extracting of the phenotype-specific sequences that positively BLAST-ed against ISMbovs, VSPs, and HAD-family hydrolases from individually reference-assembled and *de novo* assembled genomes will provide more support for their presence in particular phenotypes.

7.2 Conclusion

Mycoplasma bovis datasets were created from sequenced laboratory-grown cultures of *M. bovis* and consisted *in vitro* mixes of isolates and of *in silico* mixes of read sets. These datasets were developed and used to assist in the construction of the SepSIS pipeline. The pipeline is available at "`https://github.com/MatthewWaldner/sepsis`". SepSIS attempts to identify sequences that are specific to individual strains within a sequenced *M. bovis* 'isolate' containing multiple strains. SepSIS was evaluated through comparison of sequences produced by blind methods that rely on coverage levels against a method of SepSIS developed to use metadata to assign strain-specificity to subsequences within an assembly graph created by SPAdes. The coverage-based modes were unsuccessful in reliably replicating the results of the metadata-based validation method. In addition, the validation method of SepSIS seems to become more unreliable when larger synthetic mixes of strains are used, owing in part to the functional limitations of SPAdes and minimap2 on such datasets. However, there were positive results from SepSIS. Possible contamination with *S. maltophilia* was deemed to have been adequately removed with early post-processing steps, and a single isolate that did contain possible contaminates after the early steps had the contaminate sequences removed. Intentional contamination of *in silico* mixes showed that the post-processing steps of SepSIS adequately remove contaminating sequences as well. The metadata-based validation method was used independently to conduct two additional investigations. The first concluded that there is evidence to suggest that multiple genetically distinct strains may exist on single culture plate at one time. In the second, *M. bovis* sequences assessed as specific to lung and stifle joint tissue tropisms were identified, with ISMbov, VSPs, and HAD-family hydrolases being the most commonly identified sequences. These results show the usefulness of SepSIS when given a synthetically-mixed dataset and used to produce strain-specific sequences that describe a phenotype.

References

[1] M Aebi, M Bodmer, J Frey, and P Pilo. Herd-specific strains of Mycoplasma bovis in outbreaks of mycoplasmal mastitis and pneumonia. *Veterinary Microbiology*, 157(3-4):363–368, 2012.

[2] D Aird, MG Ross, W Chen, M Danielsson, T Fennell, C Russ, DB Jaffe, C Nusbaum, and A Gnirke. Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome biology*, 12(2):R18, 2011.

[3] SF Altschul, W Gish, W Miller, EW Myers, and DJ Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[4] JA Baaijens, AZ El Aabidine, E Rivals, and A Schönhuth. De novo assembly of viral quasispecies using overlap graphs. *Genome Research*, 27(5):835–848, 2017.

[5] A Bankevich, S Nurk, D Antipov, AA Gurevich, M Dvorkin, AS Kulikov, VM Lesin, SI Nikolenko, S Pham, AD Prjibelski, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, 19(5):455–477, 2012.

[6] AM Bolger, M Lohse, and B Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014.

[7] DR Brown, M May, JM Bradbury, MF Balish, MJ Calcutt, JI Glass, S Tasker, JB Messick, K Johansson, and H Neimark. Mycoplasma. *Bergey's Manual of Systematics of Archaea and Bacteria*, pages 1–78, 2015.

[8] J Butler, I MacCallum, M Kleber, IA Shlyakhter, MK Belmonte, ES Lander, C Nusbaum, and DB Jaffe. ALLPATHS: de novo assembly of whole-genome shotgun microreads. *Genome Research*, 18(5):810–820, 2008.

[9] C Camacho, G Coulouris, V Avagyan, N Ma, J Papadopoulos, K Bealer, and TL Madden. BLAST+: architecture and applications. *BMC Bioinformatics*, 10(1):421, 2009.

[10] ST Cowan. Principles and practice of bacterial taxonomy—a forward look. *Microbiology*, 39(1):143–153, 1965.

[11] L Dijkshoorn, BM Ursing, and JB Ursing. Strain, clone and species: comments on three basic concepts of bacteriology. *Journal of Medical Microbiology*, 49(5):397–401, 2000.

[12] NW Dyer, DF Krogh, and LP Schaan. Pulmonary mycoplasmosis in farmed white-tailed deer (Odocoileus virginianus). *Journal of Wildlife Diseases*, 40(2):366–370, 2004.

[13] MiSeq Genomics. Whole-genome Re-sequencing. `http://www.genomics.hk/PlantWhole.html`.

[14] Illumina. Illumina Sequencing by Synthesis. `https://www.youtube.com/watch?v=fCd6B5HRaZ8`, 2016.

[15] JM Janda and SL Abbott. 16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: pluses, perils, and pitfalls. *Journal of Clinical Microbiology*, 45(9):2761–2764, 2007.

[16] KK Johnson and DL Pendell. Market impacts of reducing the prevalence of bovine respiratory disease in United States beef cattle feedlots. *Frontiers in Veterinary Science*, 4:189, 2017.

[17] C Josi, S Bürki, S Vidal, E Dordet-Frisoni, C Citti, L Falquet, and P Pilo. Large-Scale Analysis of the Mycoplasma bovis Genome Identified Non-essential, Adhesion-and Virulence-Related Genes. *Frontiers in Microbiology*, 10:2085, 2019.

[18] A Justice-Allen, J Trujillo, R Corbett, R Harding, G Goodell, and D Wilson. Survival and replication of Mycoplasma species in recycled bedding sand and association with mastitis on dairy farms in Utah. *Journal of Dairy Science*, 93(1):192–202, 2010.

[19] KT Konstantinidis and JM Tiedje. Genomic insights that advance the species definition for prokaryotes. *Proceedings of the National Academy of Sciences*, 102(7):2567–2572, 2005.

[20] B Langmead and SL Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357, 2012.

[21] H Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv:1303.3997*, 2013.

[22] H Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018.

[23] H Li, B Handsaker, A Wysoker, T Fennell, J Ruan, N Homer, G Marth, G Abecasis, and R Durbin. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[24] I Lysnyansky and RD Ayling. Mycoplasma bovis: mechanisms of resistance and trends in antimicrobial susceptibility. *Frontiers in Microbiology*, 7:595, 2016.

[25] MCJ Maiden, JA Bygraves, E Feil, G Morelli, JE Russell, R Urwin, Q Zhang, J Zhou, K Zurth, DA Caugant, et al. Multilocus sequence typing: a portable approach to the identification of clones within populations of pathogenic microorganisms. *Proceedings of the National Academy of Sciences*, 95(6):3140–3145, 1998.

[26] FP Maunsell, AR Woolums, D Francoz, RF Rosenbusch, DL Step, David J Wilson, and ED Janzen. Mycoplasma bovis infections in cattle. *Journal of Veterinary Internal Medicine*, 25(4):772–783, 2011.

[27] RAJ Nicholas and RD Ayling. Mycoplasma bovis: disease, diagnosis, and control. *Research in Veterinary Science*, 74(2):105–112, 2003.

[28] Sergey Nurk, Dmitry Meleshko, Anton Korobeynikov, and Pavel A Pevzner. metaspades: a new versatile metagenomic assembler. *Genome research*, 27(5):824–834, 2017.

[29] H Ongor, R Kalin, M Karahan, B Cetinkaya, L McAuliffe, and RAJ Nicholas. Isolation of Mycoplasma bovis from broiler chickens in Turkey. *Avian Pathology*, 37(6):587–588, 2008.

[30] AM Parker, PA Sheehy, MS Hazelton, KL Bosward, and JK House. A review of mycoplasma diagnostics in cattle. *Journal of Veterinary Internal Medicine*, 32(3):1241–1252, 2018.

[31] M Patterson. NZ First welcomes MPI apology for M. Bovis failings. *New Zealand First*, Jul 2019.

[32] KD Pruitt, T Tatusova, and DR Maglott. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, 35(suppl_1):D61–D65, 2006.

[33] S Pulido-Tamayo, A Sánchez-Rodríguez, T Swings, B Van den Bergh, A Dubey, H Steenackers, J Michiels, J Fostier, and K Marchal. Frequency-based haplotype reconstruction from deep sequencing data of bacterial populations. *Nucleic Acids Research*, 43(16):e105–e105, 2015.

[34] MA Quail, M Smith, P Coupland, TD Otto, SR Harris, TR Connor, A Bertoni, HP Swerdlow, and Y Gu. A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genomics*, 13(1):341, 2012.

[35] DA Rasko, PL Worsham, TG Abshire, ST Stanley, JD Bannan, MR Wilson, RJ Langham, RS Decker, L Jiang, TD Read, et al. Bacillus anthracis comparative genome analysis in support of the Amerithrax investigation. *Proceedings of the National Academy of Sciences*, 108(12):5027–5032, 2011.

[36] KB Register, I Lysnyansky, MD Jelinski, WD Boatwright, M Waldner, DO Bayles, P Pilo, and DP Alt. Comparison of Two Multilocus Sequence Typing Schemes for Mycoplasma bovis and Revision of the PubMLST Reference Method. *Journal of Clinical Microbiology*, 58(6), 2020.

[37] KB Register, L Thole, RF Rosenbush, and FC Minion. Multilocus sequence typing of Mycoplasma bovis reveals host-specific genotypes in cattle versus bison. *Veterinary Microbiology*, 175(1):92–98, 2015.

[38] M Roosaare, M Vaher, L Kaplinski, M Möls, R Andreson, M Lepamets, T Koressaar, P Naaber, S Koljalg, and M Remm. StrainSeeker: fast identification of bacterial strains from raw sequencing reads using user-provided guide trees. *PeerJ*, 5:e3353, 2017.

[39] RF Rosenbusch, JM Kinyon, M Apley, ND Funk, S Smith, and LJ Hoffman. In vitro antimicrobial inhibition profiles of Mycoplasma bovis isolates recovered from various regions of the United States from 2002 to 2003. *Journal of Veterinary Diagnostic Investigation*, 17(5):436–441, 2005.

[40] R Rossell'o-Mora and R Amann. The species concept for prokaryotes. *FEMS microbiology reviews*, 25(1):39–67, 2001.

[41] R Rozov, A Brown Kav, David Bogumil, Naama Shterzer, Eran Halperin, Itzhak Mizrahi, and Ron Shamir. Recycler: an algorithm for detecting plasmids from de novo assembly graphs. *Bioinformatics*, 33(4):475–482, 2017.

[42] K Sachse, JH Helbig, I Lysnyansky, C Grajetzki, W Müller, E Jacobs, and D Yogev. Epitope mapping of immunogenic and adhesive structures in repetitive domains of Mycoplasma bovis variable surface lipoproteins. *Infection and Immunity*, 68(2):680–687, 2000.

[43] K Sachse, H Pfützner, M Heller, and I Hänel. Inhibition of Mycoplasma bovis cytadherence by a monoclonal antibody and various carbohydrate substances. *Veterinary Microbiology*, 36(3-4):307–316, 1993.

[44] EW Sayers, J Beck, JR Brister, EE Bolton, K Canese, DC Comeau, K Funk, A Ketter, S Kim, A Kimchi, et al. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 48(D1):D9, 2020.

[45] PM Shrestha, KP Nevin, M Shrestha, and DR Lovley. When is a microbial culture "pure"? persistent cryptic contaminant escapes detection even with deep genome sequencing. *MBio*, 4(2):e00591–12, 2013.

[46] TX: StataCorp LLC. StataCorp, College Station. Stata Statistical Software: Release 15, 2017.

[47] J Vandecraen, M Chandler, A Aertsen, and R Van Houdt. The impact of insertion sequences on bacterial genome plasticity and adaptability. *Critical Reviews in Microbiology*, 43(6):709–730, 2017.

[48] EM Vilei, I Correia, MH Ferronha, DF Bischof, and J Frey. β-D-Glucoside utilization by Mycoplasma mycoides subsp. mycoides SC: possible involvement in the control of cytotoxicity towards bovine lung cells. *BMC Microbiology*, 7(1):31, 2007.

[49] LG Wayne, DJ Brenner, RR Colwell, PAD Grimont, O Kandler, MI Krichevsky, LH Moore, WEC Moore, RGE Murray, ESMP Stackebrandt, et al. Report of the ad hoc committee on reconciliation of approaches to bacterial systematics. *International Journal of Systematic and Evolutionary Microbiology*, 37(4):463–464, 1987.

[50] KS Wise, MJ Calcutt, MF Foecking, K Röske, R Madupu, and BA Methé. Complete genome sequence of Mycoplasma bovis type strain PG45 (ATCC 25523). *Infection and Immunity*, 79(2):982–983, 2011.

[51] CR Woese, E Stackebrandt, TJ Macke, and GE Fox. A phylogenetic definition of the major eubacterial taxa. *Systematic and Applied Microbiology*, 6:143–151, 1985.

[52] DE Wood and SL Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3):R46, 2014.

[53] M Zaharia, WJ Bolosky, K Curtis, A Fox, D Patterson, S Shenker, I Stoica, RM Karp, and T Sittler. Faster and more accurate sequence alignment with SNAP. *arXiv preprint arXiv:1111.5572*, 2011.

[54] Daniel R Zeigler. Gene sequences useful for predicting relatedness of whole genomes in bacteria. *International journal of systematic and evolutionary microbiology*, 53(6):1893–1900, 2003.

Appendix A

Supplemental Tables

The tables in this Appendix contain supplemental information for the isolates used in this thesis and the results presented in Sections 5.1 and 5.5. Table A.1 contains the metadata of all of the isolates used in this thesis and Table A.2 contains the assembly statistics for independent *de novo* and reference assembly of each of the isolate. Tables A.3 to A.9 contain the processed results from the SepSIS runs presented in Section 5.1. Each of these tables contain the results for a single set of mixes. The sets of mixes are presented in Section 4.5.4. Tables A.10 to A.16 contain the raw data used to calculate the results shown in Tables A.3 to A.9, in respective order. Tables A.17 and A.18 contain the Lung-Specific-Tropism Sequences and the Stifle-Specific-Tropism Sequences that are discussed in Section 5.5.

Table A.1: The metadata of the isolates used in the thesis. The location information has been obfuscated for the sake of confidentiality.

ID	Read Set ID	Species	Sampled Tissue Type	Sampling Date	Location
MPLM_7	C	Mycoplasma bovis	Lung	2016-Dec-01	Feedlot B
MPLM_8	C	Mycoplasma bovis	Patella	2016-Dec-01	Feedlot B
MPLM_9	C	Mycoplasma bovis	Lung	2016-Dec-09	Feedlot B
MPLM_10	C	Mycoplasma bovis	Patella	2016-Dec-09	Feedlot B
MPLM_11	C	Mycoplasma bovis	Lung	2016-Dec-29	Feedlot B
MPLM_12	C	Mycoplasma bovis	Patella	2016-Dec-29	Feedlot B
MPLM_15	C	Mycoplasma bovis	Lung	2016-Dec-30	Feedlot B
MPLM_16	C	Mycoplasma bovis	Patella	2016-Dec-30	Feedlot B
MPLM_17.1	C	Mycoplasma bovis	Lung	2016-Dec-29	Feedlot B
MPLM_18.1	C	Mycoplasma bovis	Patella	2016-Dec-29	Feedlot B
MPLM_19.1	C	Mycoplasma bovis	Lung	2016-Dec-13	Feedlot B
MPLM_20.1	C	Mycoplasma bovis	Patella	2016-Dec-13	Feedlot B
MPLM_25.1	C	Mycoplasma bovis	Lung	2016-Dec-15	Feedlot B
MPLM_26.1	C	Mycoplasma bovis	Patella	2016-Dec-15	Feedlot B
MPLM_29.1	C	Mycoplasma bovis	Lung	2016-Dec-15	Feedlot B
MPLM_30.1	C	Mycoplasma bovis	Patella	2016-Dec-15	Feedlot B
MPLM_35	C	Mycoplasma bovis	Lung	2016-Dec-19	Feedlot B
MPLM_36	C	Mycoplasma bovis	Patella	2016-Dec-19	Feedlot B
MPLM_37.1	C	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MPLM_38.1	C	Mycoplasma bovis	Patella	2017-Jan-24	Feedlot B
MPLM_39	C	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MPLM_40	C	Mycoplasma bovis	Patella	2017-Jan-24	Feedlot B
MPLM_54	C	Mycoplasma bovis	Lung	2017-Dec-12	Feedlot A
MPLM_55	C	Mycoplasma bovis	Patella	2017-Dec-12	Feedlot A
MPLM_57	C	Mycoplasma bovis	Lung	2018-Jan-03	Feedlot A
MPLM_58	C	Mycoplasma bovis	Patella	2018-Jan-03	Feedlot A
MPLM_60	C	Mycoplasma bovis	Lung	2018-Jan-03	Feedlot A
MPLM_61	C	Mycoplasma bovis	Patella	2018-Jan-03	Feedlot A
MPLM_63	C	Mycoplasma bovis	Lung	2018-Jan-10	Feedlot A
MPLM_64	C	Mycoplasma bovis	Patella	2018-Jan-10	Feedlot A
MPLM_93.5	C	Mycoplasma bovis	Lung	2018-Jan-31	Feedlot A
MPLM_94.4	C	Mycoplasma bovis	Patella	2018-Jan-31	Feedlot A

MPLM_96.4	C	Mycoplasma bovis	Lung	2018-Jan-31	Feedlot A
MPLM_97.5	C	Mycoplasma bovis	Patella	2018-Jan-31	Feedlot A
MPLM_99.2	C	Mycoplasma bovis	Lung	2018-Jan-31	Feedlot A
MPLM_100.5	C	Mycoplasma bovis	Patella	2018-Jan-31	Feedlot A
MPLM_102.4	C	Mycoplasma bovis	Lung	2018-Feb-07	Feedlot A
MPLM_103.4	C	Mycoplasma bovis	Patella	2018-Feb-07	Feedlot A
MPLM_105.4	C	Mycoplasma bovis	Lung	2018-Feb-07	Feedlot A
MPLM_106.5	C	Mycoplasma bovis	Patella	2018-Feb-07	Feedlot A
MPLM_108.1	C	Mycoplasma bovis	Lung	2018-Feb-07	Feedlot A
MPLM_109.5	C	Mycoplasma bovis	Patella	2018-Feb-07	Feedlot A
MPLM_111.2	C	Mycoplasma bovis	Lung	2018-Feb-07	Feedlot A
MPLM_112.2	C	Mycoplasma bovis	Patella	2018-Feb-07	Feedlot A
MPLM_114.2	C	Mycoplasma bovis	Lung	2018-Feb-07	Feedlot A
MPLM_115.1	C	Mycoplasma bovis	Patella	2018-Feb-07	Feedlot A
MPLM_117.1	C	Mycoplasma bovis	Lung	2018-Feb-07	Feedlot A
MPLM_118.5	C	Mycoplasma bovis	Patella	2018-Feb-07	Feedlot A
MPLM_45.1	B	Mycoplasma bovis	Lung	2017-Oct-03	Feedlot A
MPLM_46.1	B	Mycoplasma bovis	Patella	2017-Oct-03	Feedlot A
MPLM_5.1	B	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot A
MPLM_6.1	B	Mycoplasma bovis	Patella	2017-Jan-24	Feedlot A
MPLM_90.1	B	Mycoplasma bovis	Lung	2018-Jan-24	Feedlot A
MPLM_91.1	B	Mycoplasma bovis	Patella	2018-Jan-24	Feedlot A
MJ121	A	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MJ126	A	Mycoplasma bovis	Patella	2017-Jan-24	Feedlot B
MJ131	A	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MJ136	A	Mycoplasma bovis	Patella	2017-Jan-24	Feedlot B
MYCO88_U	F	Mycoplasma bovis	Unspecified Tissue	2015-Feb-01	Feedlot C
MYCO86_U	F	Mycoplasma bovis	Unspecified Tissue	2016-Feb-01	Feedlot F
MP0006_SUK	G	Mycoplasma bovis	Unspecified Tissue	2015-Oct-20	Feedlot G
MP0004_TUD	G	Mycoplasma bovis	Unspecified Tissue	2015-Oct-19	Feedlot G
MYCO35_TUD	F	Mycoplasma bovis	Unspecified Tissue	N/A	N/A
MYCO81_UJ	F	Mycoplasma bovis	Unspecified Tissue	2014-Dec-01	Feedlot E
mix_sample_1_2	E	Mycoplasma bovis	Mixture	Mixture	Mixture
mix_sample_1_3	E	Mycoplasma bovis	Mixture	Mixture	Mixture
mix_sample_2_3	E	Mycoplasma bovis	Mixture	Mixture	Mixture
mix_sample_1_2_3	E	Mycoplasma bovis	Mixture	Mixture	Mixture
mix_sample_1_3_5	E	Mycoplasma bovis	Mixture	Mixture	Mixture
mix_sample_3_4_5	E	Mycoplasma bovis	Mixture	Mixture	Mixture
mix_sample_1_2_3_4	E	Mycoplasma bovis	Mixture	Mixture	Mixture
mix_sample_1_2_3_5	E	Mycoplasma bovis	Mixture	Mixture	Mixture
mix_sample_1_2_3_4_5	E	Mycoplasma bovis	Mixture	Mixture	Mixture
mix_sample_1_2_3_5_6	E	Mycoplasma bovis	Mixture	Mixture	Mixture
MJ122	A	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MJ123	A	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MJ124	A	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MJ125	A	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MJ127	A	Mycoplasma bovis	Joint	2017-Jan-24	Feedlot B
MJ128	A	Mycoplasma bovis	Joint	2017-Jan-24	Feedlot B
MJ129	A	Mycoplasma bovis	Joint	2017-Jan-24	Feedlot B
MJ130	A	Mycoplasma bovis	Joint	2017-Jan-24	Feedlot B
MJ132	A	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B

MJ133	A	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MJ134	A	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MJ135	A	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot B
MJ137	A	Mycoplasma bovis	Joint	2017-Jan-24	Feedlot B
MJ138	A	Mycoplasma bovis	Joint	2017-Jan-24	Feedlot B
MJ139	A	Mycoplasma bovis	Joint	2017-Jan-24	Feedlot B
MJ140	A	Mycoplasma bovis	Joint	2017-Jan-24	Feedlot B
MPLM_45.2	B	Mycoplasma bovis	Lung	2017-Oct-03	Feedlot A
MPLM_46.2	B	Mycoplasma bovis	Patella	2017-Oct-03	Feedlot A
MPLM_5.2	B	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot A
MPLM_6.2	B	Mycoplasma bovis	Patella	2017-Jan-24	Feedlot A
MPLM_90.2	B	Mycoplasma bovis	Lung	2018-Jan-24	Feedlot A
MPLM_91.2	B	Mycoplasma bovis	Patella	2018-Jan-24	Feedlot A
MPLM_45.3	B	Mycoplasma bovis	Lung	2017-Oct-03	Feedlot A
MPLM_46.3	B	Mycoplasma bovis	Patella	2017-Oct-03	Feedlot A
MPLM_5.3	B	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot A
MPLM_6.3	B	Mycoplasma bovis	Patella	2017-Jan-24	Feedlot A
MPLM_90.3	B	Mycoplasma bovis	Lung	2018-Jan-24	Feedlot A
MPLM_91.3	B	Mycoplasma bovis	Patella	2018-Jan-24	Feedlot A
MPLM_45.4	B	Mycoplasma bovis	Lung	2017-Oct-03	Feedlot A
MPLM_46.4	B	Mycoplasma bovis	Patella	2017-Oct-03	Feedlot A
MPLM_5.4	B	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot A
MPLM_6.4	B	Mycoplasma bovis	Patella	2017-Jan-24	Feedlot A
MPLM_90.4	B	Mycoplasma bovis	Lung	2018-Jan-24	Feedlot A
MPLM_91.4	B	Mycoplasma bovis	Patella	2018-Jan-24	Feedlot A
MPLM_45.5	B	Mycoplasma bovis	Lung	2017-Oct-03	Feedlot A
MPLM_46.5	B	Mycoplasma bovis	Patella	2017-Oct-03	Feedlot A
MPLM_5.5	B	Mycoplasma bovis	Lung	2017-Jan-24	Feedlot A
MPLM_6.5	B	Mycoplasma bovis	Patella	2017-Jan-24	Feedlot A
MPLM_90.5	B	Mycoplasma bovis	Lung	2018-Jan-24	Feedlot A
MPLM_91.5	B	Mycoplasma bovis	Patella	2018-Jan-24	Feedlot A
MPLM_212.2	D	Mycoplasma agalactiae	Nasopharyngeal Swab	2009-Mar-21	Feedlot D
MP00_57	D	Mycoplasma arginini	Unspecified Tissue	2015-Dec-05	Feedlot E
MP00_211	D	Mycoplasma bovirhinis	Unspecified Tissue	2015-Jan-01	Feedlot H

Table A.2: The assembly statistics of the isolates used in the thesis. These statistics are for individual *de novo* and reference assemblies of the isolates. This is given as a measure of quality for the individual isolates used in this thesis.

ID	Paired Read Count	Coverage Depth	Reference Assembly Length	Contigs Total Length	Contigs Number	N50	NG50
MPLM_7	15510	147.2879222	1003404	4213354	4703	928	34286
MPLM_8	24529	236.689437	1003404	3680674	1463	8382	31646
MPLM_9	15833	168.4677246	1003404	1522078	1205	16641	27434
MPLM_10	3976	25.54332456	1003404	1039242	326	28856	28856
MPLM_11	2429	18.77520529	1003361	988652	437	8038	7794
MPLM_12	3147	26.30272189	1003302	836226	402	9387	6879

MPLM_15	15534	161.5802101	1003404	1474351	1111	16802	24557
MPLM_16	15337	163.7826858	1003404	4574925	517	69293	239526
MPLM_17.1	14100	110.9573206	1003394	8733721	4151	3843	26839
MPLM_18.1	8268	74.65762371	1003389	5433419	2128	12464	35610
MPLM_19.1	19733	148.3533202	1003404	6066905	3161	19690	73660
MPLM_20.1	738	10.2397348	1002738	515773	976	491	384
MPLM_25.1	9406	55.64230457	1003324	6228601	3402	5386	21463
MPLM_26.1	9227	73.41306786	1003328	7187222	4632	2738	22971
MPLM_29.1	16392	173.9705421	1003403	10574678	4229	6275	67258
MPLM_30.1	210	1.669159601	1002471	23900	54	423	0
MPLM_35	4206	40.25227647	1003404	1159893	578	10029	12894
MPLM_36	9168	67.8545555	1003394	4387625	3456	1621	16091
MPLM_37.1	1088	10.41726066	1003341	1383375	1702	905	1138
MPLM_38.1	10623	76.93748832	1003371	9536324	5283	3474	28930
MPLM_39	15785	209.6448264	1003404	2360890	2377	1048	29681
MPLM_40	7428	54.66174314	1003404	4093928	3440	1436	14467
MPLM_54	10508	92.95850233	1003302	7131568	5362	2111	8932
MPLM_55	5870	40.93390496	1003404	3678763	3505	1231	16226
MPLM_57	11577	106.1599964	1003404	3621430	3331	1308	22052
MPLM_58	8363	89.90532102	1003404	3458747	1149	6112	31354
MPLM_60	2292	18.16539681	1003342	1641282	1596	1433	3363
MPLM_61	10685	97.90191452	1003404	3586314	1383	8194	31656
MPLM_63	14211	174.5389676	1003404	3413721	1317	5028	33328
MPLM_64	3251	28.74214357	1003388	1186381	660	14698	16697
MPLM_93.5	24479	275.0807578	1003404	7910509	3756	23910	108018
MPLM_94.4	7284	47.39749139	1003404	2925193	733	124310	225465
MPLM_96.4	11397	89.82797417	1003302	7406449	3836	19019	90112
MPLM_97.5	13415	91.21793439	1003298	5425168	997	34375	97435
MPLM_99.2	4409	29.42538339	1002738	4315696	2481	2687	7802
MPLM_100.5	1406	12.13308472	981803	1034804	1994	497	502
MPLM_102.4	16607	126.6068582	1003398	6368241	2497	24534	80666
MPLM_103.4	19206	225.1158153	1003404	4742367	1126	27895	82790
MPLM_105.4	1271	9.329898735	995630	625283	1238	479	409
MPLM_106.5	929	8.805489686	1001262	429190	725	514	0
MPLM_108.1	4792	28.414953	1002014	3155384	2192	2279	4712
MPLM_109.5	15820	110.8935359	1003364	4670350	1128	25023	68233
MPLM_111.2	9027	64.43139795	1003404	6889702	6519	1344	7159
MPLM_112.2	19409	153.9114578	1003346	4581155	1198	17779	57114
MPLM_114.2	12461	105.4624998	1003371	4751088	1164	22959	59104
MPLM_115.1	19086	156.6293922	1003394	9150273	4793	7820	78461
MPLM_117.1	4001	31.16070317	1003331	5133269	5998	966	2527
MPLM_118.5	4742	31.18267692	1002145	4017691	1875	3871	10428
MPLM_45.1	22368	172.2948469	1003309	5281362	873	60694	125776
MPLM_46.1	9496	93.11082721	1003380	8950087	3186	5438	41885
MPLM_5.1	20732	198.2339846	1003404	3256081	719	79758	225137
MPLM_6.1	1871	19.14810132	1003404	1059885	571	5489	5861
MPLM_90.1	19341	246.0517018	1003404	4329399	3045	4578	159277
MPLM_91.1	1659	18.82621868	1003404	987624	796	1990	1975
MJ121	22199	124.699296	1003404	978344	230	25712	21937
MJ126	27545	194.6996675	1003404	975431	254	27596	26516
MJ131	56226	549.6667942	1003404	1001210	232	23960	23960

MJ136	40304	366.5318574	1003404	1079874	417	20788	25180
MYCO88_U	6667	53.84614699	1003404	960056	233	21621	20804
MYCO86_U	9074	67.01421966	1003404	962834	247	25051	25051
MYCO35_TUD	8001	44.75674251	1003404	955600	173	27541	27541
MYCO81_UJ	18242	139.4921974	1003404	959661	339	7207	7074
mix_sample_1_2	12603	87.40717362	1003404	11499251	5446	5157	46046
mix_sample_1_3	17431	134.3426488	1003272	12007300	5653	8533	71764
mix_sample_2_3	976	6.41890457	1003167	484358	773	635	0
mix_sample_1_2_3	6914	45.34144376	1003069	6151098	6031	1341	5642
mix_sample_1_3_5	4964	42.86992784	1003273	6048917	7932	826	2872
mix_sample_3_4_5	3238	27.7003037	1003404	2942243	4352	670	1660
mix_sample_1_2_3_4	5482	48.55294164	1002864	9183153	6279	3049	30902
mix_sample_1_2_3_5	7944	46.39309528	1003404	11351253	9721	1710	27271
mix_sample_1_2_3_4_5	1829	13.29386503	1003272	649610	1295	466	410
mix_sample_1_2_3_5_6	12824	120.6870867	1003404	11818327	11005	1418	34146
MJ122	9630	51.04471062	1003404	973961	231	23028	23028
MJ123	22782	102.312981	1003404	979164	256	26515	25185
MJ124	21659	188.1769685	1003404	968832	222	21701	20813
MJ125	35661	333.0276014	1003404	981486	260	21902	21902
MJ127	39700	256.5380334	1003404	972016	227	24970	22686
MJ128	21565	141.1526976	1003404	971705	221	24889	24889
MJ129	31493	159.1881438	1003404	973538	230	27596	27596
MJ130	34552	329.4627467	1003404	972509	237	26516	24970
MJ132	33552	268.0456814	1003404	1006937	272	23502	23502
MJ133	53423	322.951073	1003404	3865744	5599	731	25918
MJ134	45269	290.7326082	1003404	997231	218	25918	25918
MJ135	37113	181.0073694	1003404	1002216	237	25918	25181
MJ137	45449	419.3569583	1003404	1320097	682	5321	9225
MJ138	31261	230.1634056	1003404	1147615	602	18624	27215
MJ139	63769	290.4168881	1003404	1174364	373	9091	11900
MJ140	43403	215.543485	1003404	1593028	1503	2130	3860
MPLM_45.2	1746	12.15569902	1002877	1167603	2087	533	578
MPLM_46.2	10646	80.68671893	1003388	9697235	2702	8921	43286
MPLM_5.2	11422	113.3276556	1003404	1166209	529	23427	28326
MPLM_6.2	5245	47.27106755	1003404	2070130	1615	1998	20413
MPLM_90.2	996	6.131095086	998418	431327	881	462	0
MPLM_91.2	19805	164.1630592	1003404	3134662	1638	3150	31602
MPLM_45.3	4044	42.22534428	1003404	5665075	2077	9389	42506
MPLM_46.3	8616	53.40960486	1003319	7760080	4468	2579	8211
MPLM_5.3	21632	225.9844443	1003404	4922935	3225	3416	31806
MPLM_6.3	12658	100.4274052	1003404	3277163	1978	2499	28471
MPLM_90.3	7608	52.41149746	1003404	1099303	456	20166	22864
MPLM_91.3	9595	83.10135737	1003404	2683745	2110	1670	29254
MPLM_45.4	16602	95.73050063	1003325	9511650	2577	14333	97838
MPLM_46.4	18242	179.0143603	1002938	4959864	490	45109	100845
MPLM_5.4	16061	181.4078334	1003404	1184101	513	21041	25918
MPLM_6.4	2495	21.13422785	1003404	971360	254	14061	13063
MPLM_90.4	13149	99.33878074	1003404	2712633	2247	1654	21398
MPLM_91.4	7589	67.92264687	1003404	5101832	2042	4107	33257
MPLM_45.5	1733	15.83563645	1003162	2353333	2952	895	1633
MPLM_46.5	4	0.006870612	0	0	0	0	0

MPLM_5.5	26514	250.1308824	1003404	3188508	2810	1403	28326
MPLM_6.5	13890	119.0991685	1003404	9675916	8448	1610	24557
MPLM_90.5	12248	95.08237677	1003404	1256662	752	27594	31376
MPLM_91.5	10773	82.57384684	1003404	5851529	2535	12949	36118
MPLM_212.2	10227	79.90888474	1003246	5317124	1103	20729	74936
MP00_57	12308	41.88380725	1002020	3866636	3410	1443	4311
MP00_211	13823	98.19860597	949395	4827963	257	41350	89898

Table A.3: The coverage-based SepSIS run parameters and resulting sensitivities, positive predictive values, and number of mixes in the run for The Set of *in silico* Mixes. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	Sensitivity	Sens. 95% CI	PPV	PPV 95% CI	# of Non-Zero Strain-Specific Sequence Mixes	Total Number of Mixes
in silico	ORGANIC P	CYCLIC	10	20	0.1259	0.0847 and 0.1830	0.1180	0.0806 and 0.1696	10	10
in silico	ORGANIC P	CYCLIC	10	30	0.2752	0.2130 and 0.3476	0.1169	0.0855 and 0.1577	10	10
in silico	ORGANIC Z	CYCLIC	-1.282	-0.842	0	N/A	0	N/A	10	10
in silico	ORGANIC Z	CYCLIC	-1.282	-0.524	0.0194	0.0026 and 0.1314	0.1818	- and -	10	10
in silico	ORGANIC P	ISOLATED	10	30	0.1463	0.0141 and 0.6734	0.1429	0.0121 and 0.6946	4	10
in silico	ORGANIC P	ISOLATED	10	35	0.2927	0.0533 and 0.7525	0.2000	0.0967 and 0.5841	4	10
in silico	ORGANIC Z	ISOLATED	-1.282	-0.524	0	N/A	0	N/A	4	10
in silico	ORGANIC Z	ISOLATED	-1.282	-0.385	0	N/A	0	N/A	4	10
in silico	ORGANIC P	BOTH	10	30	0.2442	0.1886 and 0.3098	0.1082	0.0791 and 0.1713	10	10
in silico	ORGANIC P	BOTH	10	35	0.3103	0.2512 and 0.3762	0.1091	0.0850 and 0.1388	10	10
in silico	ORGANIC Z	BOTH	-1.282	-0.524	0.0027	0.0004 and 0.0196	0.1111	- and -	10	10
in silico	ORGANIC Z	BOTH	-1.282	-0.385	0.1740	0.0959 and 0.2949	0.0634	0.0444 and 0.0897	10	10

Table A.4: The coverage-based SepSIS run parameters and resulting sensitivities, positive predictive values, and number of mixes in the run for The Set of *in vitro* Mixes. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	Sensitivity	Sens. 95% CI	PPV	PPV 95% CI	# of Non-Zero Strain-Specific Sequence Mixes	Total Number of Mixes
in vitro	ORGANIC P	CYCLIC	10	20	0	N/A	0	N/A	3	10
in vitro	ORGANIC P	CYCLIC	10	30	0	N/A	0	N/A	3	10
in vitro	ORGANIC Z	CYCLIC	-1.282	-0.842	0	N/A	0	N/A	2	10
in vitro	ORGANIC Z	CYCLIC	-1.282	-0.524	0	N/A	0	N/A	3	10
in vitro	ORGANIC P	ISOLATED	10	30	0	N/A	0	N/A	2	10
in vitro	ORGANIC P	ISOLATED	10	35	0	N/A	0	N/A	2	10
in vitro	ORGANIC Z	ISOLATED	-1.282	-0.524	0	N/A	0	N/A	0	10
in vitro	ORGANIC Z	ISOLATED	-1.282	-0.385	0	N/A	0	N/A	0	10
in vitro	ORGANIC P	BOTH	10	30	0	N/A	0	N/A	2	10
in vitro	ORGANIC P	BOTH	10	35	0	N/A	0	N/A	3	10
in vitro	ORGANIC Z	BOTH	-1.282	-0.524	0	N/A	0	N/A	0	10
in vitro	ORGANIC Z	BOTH	-1.282	-0.385	0	N/A	0	N/A	0	10

Table A.5: The coverage-based SepSIS run parameters and resulting sensitivities, positive predictive values, and number of mixes in the run for The Set of Paired Isolates Mixes. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	Sensitivity	Sens. 95% CI	PPV	PPV 95% CI	# of Non-Zero Strain-Specific Sequence Mixes	Total Number of Mixes
Paired Isolates	ORGANIC P	CYCLIC	10	20	0.0189	0.0085 and 0.0414	0.0638	0.0274 and 0.1411	27	29
Paired Isolates	ORGANIC P	CYCLIC	10	30	0.0438	0.0240 and 0.0787	0.0587	0.0304 and 0.1102	27	29
Paired Isolates	ORGANIC Z	CYCLIC	-1.282	-0.842	0	N/A	0	N/A	27	29
Paired Isolates	ORGANIC Z	CYCLIC	-1.282	-0.524	0.0269	0.0124 and 0.0573	0.1190	0.0830 and 0.1679	27	29
Paired Isolates	ORGANIC P	ISOLATED	10	30	0	N/A	0	N/A	25	29
Paired Isolates	ORGANIC P	ISOLATED	10	35	0	N/A	0	N/A	25	29
Paired Isolates	ORGANIC Z	ISOLATED	-1.282	-0.524	0	N/A	0	N/A	25	29
Paired Isolates	ORGANIC Z	ISOLATED	-1.282	-0.385	0	N/A	0	N/A	25	29

Paired Isolates	ORGANIC P	BOTH	10	30	0.0053	0.0012 and 0.0236	0.0242	0.0035 and 0.1488	27	29
Paired Isolates	ORGANIC P	BOTH	10	35	0.0080	0.0016 and 0.0398	0.0230	0.0030 and 0.1555	27	29
Paired Isolates	ORGANIC Z	BOTH	-1.282	-0.524	0	N/A	0	N/A	27	29
Paired Isolates	ORGANIC Z	BOTH	-1.282	-0.385	0.0057	0.0008 and 0.0373	0.0442	0.0215 and 0.0889	27	29

Table A.6: The coverage-based SepSIS run parameters and resulting sensitivities, positive predictive values, and number of mixes in the run for The Set of Large Mixes Containing 2 Isolates. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	Sensitivity	Sens. 95% CI	PPV	PPV 95% CI	# of Non-Zero Strain-Specific Sequence Mixes	Total Number of Mixes
Large Mixes (2 Isolates)	ORGANIC P	CYCLIC	10	20	0.0109	0.0056 and 0.0210	0.0476	0.0231 and 0.0958	15	15
Large Mixes (2 Isolates)	ORGANIC P	CYCLIC	10	30	0.0311	0.0204 and 0.0470	0.0584	0.0312 and 0.1067	15	15
Large Mixes (2 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.842	0	N/A	0	N/A	14	15
Large Mixes (2 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.524	0.0235	0.0122 and 0.0447	0.0463	0.0229 and 0.0915	15	15
Large Mixes (2 Isolates)	ORGANIC P	ISOLATED	10	30	0.0861	0.0126 and 0.4110	0.2549	0.1548 and 0.3898	13	15
Large Mixes (2 Isolates)	ORGANIC P	ISOLATED	10	35	0.0861	0.0126 and 0.4110	0.1805	0.0583 and 0.4396	13	15
Large Mixes (2 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.524	0	N/A	0	N/A	12	15
Large Mixes (2 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.385	0	N/A	0	N/A	12	15
Large Mixes (2 Isolates)	ORGANIC P	BOTH	10	30	0.0152	0.0026 and 0.0835	0.1358	0.0358 and 0.3995	14	15
Large Mixes (2 Isolates)	ORGANIC P	BOTH	10	35	0.0194	0.0041 and 0.0860	0.1111	0.0311 and 0.3277	14	15

Large Mixes (2 Isolates)	ORGANIC Z	BOTH	-1.282	-0.524	0	N/A	0	N/A	14	15
Large Mixes (2 Isolates)	ORGANIC Z	BOTH	-1.282	-0.385	0	N/A	0	N/A	14	15

Table A.7: The coverage-based SepSIS run parameters and resulting sensitivities, positive predictive values, and number of mixes in the run for The Set of Large Mixes Containing 3 Isolates. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	Sensitivity	Sens. 95% CI	PPV	PPV 95% CI	# of Non-Zero Strain-Specific Sequence Mixes	Total Number of Mixes
Large Mixes (3 Isolates)	ORGANIC P	CYCLIC	10	20	0.0200	0.0067 and 0.0584	0.0863	0.0337 and 0.2037	9	9
Large Mixes (3 Isolates)	ORGANIC P	CYCLIC	10	30	0.0369	0.0177 and 0.0754	0.0652	0.0421 and 0.0995	9	9
Large Mixes (3 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.842	0	N/A	0	N/A	9	9
Large Mixes (3 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.524	0.0274	0.0163 and 0.0458	0.0648	0.0364 and 0.1129	9	9
Large Mixes (3 Isolates)	ORGANIC P	ISOLATED	10	30	0.1594	0.0350 and 0.4982	0.2500	0.2081 and 0.2971	9	9
Large Mixes (3 Isolates)	ORGANIC P	ISOLATED	10	35	0.1594	0.0350 and 0.4982	0.1979	0.0819 and 0.4055	9	9
Large Mixes (3 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.524	0	N/A	0	N/A	9	9
Large Mixes (3 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.385	0	N/A	0	N/A	9	9
Large Mixes (3 Isolates)	ORGANIC P	BOTH	10	30	0.0158	0.0019 and 0.1191	0.2429	0.1775 and 0.5796	9	9
Large Mixes (3 Isolates)	ORGANIC P	BOTH	10	35	0.0177	0.0021 and 0.1319	0.1979	0.0819 and 0.4055	9	9
Large Mixes (3 Isolates)	ORGANIC Z	BOTH	-1.282	-0.524	0	N/A	0	N/A	9	9

Large Mixes (3 Isolates)	ORGANIC Z	BOTH	-1.282	-0.385	0	N/A	0	N/A	9	9

Table A.8: The coverage-based SepSIS run parameters and resulting sensitivities, positive predictive values, and number of mixes in the run for The Set of Large Mixes Containing 4 Isolates. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	Sensitivity	Sens. 95% CI	PPV	PPV 95% CI	# of Non-Zero Strain-Specific Sequence Mixes	Total Number of Mixes
Large Mixes (4 Isolates)	ORGANIC P	CYCLIC	10	20	0.0358	0.0208 and 0.0610	0.0631	0.0398 and 0.0986	6	6
Large Mixes (4 Isolates)	ORGANIC P	CYCLIC	10	30	0.0728	0.0337 and 0.1502	0.1169	0.0855 and 0.1577	6	6
Large Mixes (4Strains)	ORGANIC Z	CYCLIC	-1.282	-0.842	0	N/A	0	N/A	6	6
Large Mixes (4 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.524	0.0291	0.0083 and 0.0967	0.1320	0.1034 and 0.1670	6	6
Large Mixes (4 Isolates)	ORGANIC P	ISOLATED	10	30	0.0571	0.0340 and 0.0946	0.0667	- and -	5	6
Large Mixes (4 Isolates)	ORGANIC P	ISOLATED	10	35	0.0571	0.0340 and 0.0946	0.0476	- and -	5	6
Large Mixes (4 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.524	0	N/A	0	N/A	5	6
Large Mixes (4 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.385	0	N/A	0	N/A	5	6
Large Mixes (4 Isolates)	ORGANIC P	BOTH	10	30	0.0166	0.0066 and 0.0413	0.0541	0.0406 and 0.0717	6	6
Large Mixes (4 Isolates)	ORGANIC P	BOTH	10	35	0.0197	0.0063 and 0.0594	0.0384	0.0269 and 0.0546	6	6
Large Mixes (4 Isolates)	ORGANIC Z	BOTH	-1.282	-0.524	0	N/A	0	N/A	6	6
Large Mixes (4 Isolates)	ORGANIC Z	BOTH	-1.282	-0.385	0	N/A	0	N/A	6	6

Table A.9: The coverage-based SepSIS run parameters and resulting sensitivities, positive predictive values, and number of mixes in the run for The Set of Large Mixes Containing 5 Isolates. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	Sensitivity	Sens. 95% CI	PPV	PPV 95% CI	# of Non-Zero Strain-Specific Sequence Mixes	Total Number of Mixes
Large Mixes (5 Isolates)	ORGANIC P	CYCLIC	10	20	0.0253	0.0081 and 0.0757	0.0847	0.0239 and 0.2592	3	3
Large Mixes (5 Isolates)	ORGANIC P	CYCLIC	10	30	0.0354	0.0216 and 0.0574	0.0422	0.0245 and 0.0716	3	3
Large Mixes (5 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.842	0	N/A	0	N/A	3	3
Large Mixes (5 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.524	0.0101	0.0037 and 0.0271	0.0101	0.0037 and 0.0271	3	3
Large Mixes (5 Isolates)	ORGANIC P	ISOLATED	10	30	0	N/A	0	N/A	3	3
Large Mixes (5 Isolates)	ORGANIC P	ISOLATED	10	35	0	N/A	0	N/A	3	3
Large Mixes (5 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.524	0	N/A	0	N/A	3	3
Large Mixes (5 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.385	0	N/A	0	N/A	3	3
Large Mixes (5 Isolates)	ORGANIC P	BOTH	10	30	0	N/A	0	N/A	3	3
Large Mixes (5 Isolates)	ORGANIC P	BOTH	10	35	0	N/A	0	N/A	3	3
Large Mixes (5 Isolates)	ORGANIC Z	BOTH	-1.282	-0.524	0	N/A	0	N/A	3	3
Large Mixes (5 Isolates)	ORGANIC Z	BOTH	-1.282	-0.385	0	N/A	0	N/A	3	3

Table A.10: The coverage-based SepSIS run parameters, the raw counts used to calculate sensitivities and PPVs, and number of mixes in the run for The Set of *in silico* Mixes. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	# of True Predicted Strain-Specific Sequences	# of Coverage-Based Method Predicted Sequences (PPV Denominator)	# of Verification Method Sequences (Sensitivity Denominator)	Total Number of Mixes
in silico	ORGANIC P	CYCLIC	10	20	273	2313	2169	10
in silico	ORGANIC P	CYCLIC	10	30	597	5109	2169	10
in silico	ORGANIC Z	CYCLIC	-1.282	-0.842	0	0	2169	10
in silico	ORGANIC Z	CYCLIC	-1.282	-0.524	42	231	2169	10
in silico	ORGANIC P	BOTH	10	30	543	5019	2224	10
in silico	ORGANIC P	BOTH	10	35	690	6327	2224	10
in silico	ORGANIC Z	BOTH	-1.282	-0.524	6	54	2224	10
in silico	ORGANIC Z	BOTH	-1.282	-0.385	387	6108	2224	10
in silico	ORGANIC P	ISOLATED	10	30	6	42	39	10
in silico	ORGANIC P	ISOLATED	10	35	12	60	39	10
in silico	ORGANIC Z	ISOLATED	-1.282	-0.524	0	0	39	10
in silico	ORGANIC Z	ISOLATED	-1.282	-0.385	0	0	39	10

Table A.11: The coverage-based SepSIS run parameters, the raw counts used to calculate sensitivities and PPVs, and number of mixes in the run for The Set of *in vitro* Mixes. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	# of True Predicted Strain-Specific Sequences	# of Coverage-Based Method Predicted Sequences (PPV Denominator)	# of Verification Method Sequences (Sensitivity Denominator)	Total Number of Mixes
in vitro	ORGANIC P	CYCLIC	10	20	0	6	2169	10
in vitro	ORGANIC P	CYCLIC	10	30	0	10	2169	10
in vitro	ORGANIC Z	CYCLIC	-1.282	-0.842	0	2	2169	10
in vitro	ORGANIC Z	CYCLIC	-1.282	-0.524	0	34	2169	10
in vitro	ORGANIC P	ISOLATED	10	30	0	2	39	10
in vitro	ORGANIC P	ISOLATED	10	35	0	2	39	10
in vitro	ORGANIC Z	ISOLATED	-1.282	-0.524	0	0	39	10
in vitro	ORGANIC Z	ISOLATED	-1.282	-0.385	0	0	39	10
in vitro	ORGANIC P	BOTH	10	30	0	2	2224	10
in vitro	ORGANIC P	BOTH	10	35	0	2	2224	10
in vitro	ORGANIC Z	BOTH	-1.282	-0.524	0	0	2224	10
in vitro	ORGANIC Z	BOTH	-1.282	-0.385	0	0	2224	10

Table A.12: The coverage-based SepSIS run parameters, the raw counts used to calculate sensitivities and PPVs, and number of mixes in the run for The Set of Paired Isolates Mixes. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	# of True Predicted Strain-Specific Sequences	# of Coverage-Based Method Predicted Sequences (PPV Denominator)	# of Verification Method Sequences (Sensitivity Denominator)	Total Number of Mixes
Paired Isolates	ORGANIC P	CYCLIC	10	20	38	596	2007	29
Paired Isolates	ORGANIC P	CYCLIC	10	30	88	1500	2007	29
Paired Isolates	ORGANIC Z	CYCLIC	-1.282	-0.842	0	0	2007	29
Paired Isolates	ORGANIC Z	CYCLIC	-1.282	-0.524	54	1221	2007	29
Paired Isolates	ORGANIC P	ISOLATED	10	30	2	146	531	29
Paired Isolates	ORGANIC P	ISOLATED	10	35	2	200	531	29
Paired Isolates	ORGANIC Z	ISOLATED	-1.282	-0.524	0	0	531	29
Paired Isolates	ORGANIC Z	ISOLATED	-1.282	-0.385	0	0	531	29
Paired Isolates	ORGANIC P	BOTH	10	30	14	579	2628	29
Paired Isolates	ORGANIC P	BOTH	10	35	21	912	2628	29
Paired Isolates	ORGANIC Z	BOTH	-1.282	-0.524	0	0	2628	29
Paired Isolates	ORGANIC Z	BOTH	-1.282	-0.385	15	126	2628	29

Table A.13: The coverage-based SepSIS run parameters, the raw counts used to calculate sensitivities and PPVs, and number of mixes in the run for The Set of Large Mixes Containing 2 Isolates. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	# of True Predicted Strain-Specific Sequences	# of Coverage-Based Method Predicted Sequences (PPV Denominator)	# of Verification Method Sequences (Sensitivity Denominator)	Total Number of Mixes
Large Mixes (2 Isolates)	ORGANIC P	CYCLIC	10	20	13	273	1191	15
Large Mixes (2 Isolates)	ORGANIC P	CYCLIC	10	30	37	634	1191	15
Large Mixes (2 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.842	0	12	1191	15

Large Mixes (2 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.524	28	604	1191	15
Large Mixes (2 Isolates)	ORGANIC P	ISOLATED	10	30	13	51	151	15
Large Mixes (2 Isolates)	ORGANIC P	ISOLATED	10	35	13	72	151	15
Large Mixes (2 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.524	0	6	151	15
Large Mixes (2 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.385	0	46	151	15
Large Mixes (2 Isolates)	ORGANIC P	BOTH	10	30	22	162	1445	15
Large Mixes (2 Isolates)	ORGANIC P	BOTH	10	35	28	252	1445	15
Large Mixes (2 Isolates)	ORGANIC Z	BOTH	-1.282	-0.524	0	0	1445	15
Large Mixes (2 Isolates)	ORGANIC Z	BOTH	-1.282	-0.385	0	0	1445	15

Table A.14: The coverage-based SepSIS run parameters, the raw counts used to calculate sensitivities and PPVs, and number of mixes in the run for The Set of Large Mixes Containing 3 Isolates. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	# of True Predicted Strain-Specific Sequences	# of Coverage-Based Method Predicted Sequences (PPV Denominator)	# of Verification Method Sequences (Sensitivity Denominator)	Total Number of Mixes
Large Mixes (3 Isolates)	ORGANIC P	CYCLIC	10	20	19	220	948	9
Large Mixes (3 Isolates)	ORGANIC P	CYCLIC	10	30	35	537	948	9
Large Mixes (3 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.842	0	0	948	9
Large Mixes (3 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.524	26	401	948	9

Large Mixes (3 Isolates)	ORGANIC P	ISOLATED	10	30	11	44	69	9
Large Mixes (3 Isolates)	ORGANIC P	ISOLATED	10	35	11	54	69	9
Large Mixes (3 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.524	0	0	69	9
Large Mixes (3 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.385	0	0	69	9
Large Mixes (3 Isolates)	ORGANIC P	BOTH	10	30	17	70	1076	9
Large Mixes (3 Isolates)	ORGANIC P	BOTH	10	35	19	96	1076	9
Large Mixes (3 Isolates)	ORGANIC Z	BOTH	-1.282	-0.524	0	0	1076	9
Large Mixes (3 Isolates)	ORGANIC Z	BOTH	-1.282	-0.385	0	8	1076	9

Table A.15: The coverage-based SepSIS run parameters, the raw counts used to calculate sensitivities and PPVs, and number of mixes in the run for The Set of Large Mixes Containing 4 Isolates. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	# of True Predicted Strain-Specific Sequences	# of Coverage-Based Method Predicted Sequences (PPV Denominator)	# of Verification Method Sequences (Sensitivity Denominator)	Total Number of Mixes
Large Mixes (4 Isolates)	ORGANIC P	CYCLIC	10	20	32	507	893	6
Large Mixes (4 Isolates)	ORGANIC P	CYCLIC	10	30	65	1324	893	6
Large Mixes (4 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.842	0	0	893	6
Large Mixes (4 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.524	26	197	893	6
Large Mixes (4 Isolates)	ORGANIC P	ISOLATED	10	30	2	30	35	6

Large Mixes (4 Isolates)	ORGANIC P	ISOLATED	10	35	2	42	35	6
Large Mixes (4 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.524	0	0	35	6
Large Mixes (4 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.385	0	0	35	6
Large Mixes (4 Isolates)	ORGANIC P	BOTH	10	30	16	296	964	6
Large Mixes (4 Isolates)	ORGANIC P	BOTH	10	35	19	495	964	6
Large Mixes (4 Isolates)	ORGANIC Z	BOTH	-1.282	-0.524	0	0	964	6
Large Mixes (4 Isolates)	ORGANIC Z	BOTH	-1.282	-0.385	0	0	964	6

Table A.16: The coverage-based SepSIS run parameters, the raw counts used to calculate sensitivities and PPVs, and number of mixes in the run for The Set of Large Mixes Containing 5 Isolates. These results are presented in Section 5.1 and the sets of mixes are described in Section 4.5.4.

Mix Name	RUNMODE	SUBMODE	Min Value Score	Max Value Score	# of True Predicted Strain-Specific Sequences	# of Coverage-Based Method Predicted Sequences (PPV Denominator)	# of Verification Method Sequences (Sensitivity Denominator)	Total Number of Mixes
Large Mixes (5 Isolates)	ORGANIC P	CYCLIC	10	20	5	59	198	3
Large Mixes (5 Isolates)	ORGANIC P	CYCLIC	10	30	7	166	198	3
Large Mixes (5 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.842	0	0	198	3
Large Mixes (5 Isolates)	ORGANIC Z	CYCLIC	-1.282	-0.524	2	23	198	3
Large Mixes (5 Isolates)	ORGANIC P	ISOLATED	10	30	0	0	16	3
Large Mixes (5 Isolates)	ORGANIC P	ISOLATED	10	35	0	0	16	3

Large Mixes (5 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.524	0	0	16	3
Large Mixes (5 Isolates)	ORGANIC Z	ISOLATED	-1.282	-0.385	0	0	16	3
Large Mixes (5 Isolates)	ORGANIC P	BOTH	10	30	0	0	219	3
Large Mixes (5 Isolates)	ORGANIC P	BOTH	10	35	0	0	219	3
Large Mixes (5 Isolates)	ORGANIC Z	BOTH	-1.282	-0.524	0	0	219	3
Large Mixes (5 Isolates)	ORGANIC Z	BOTH	-1.282	-0.385	0	0	219	3

Table A.17: The Lung-Specific-Tropism Sequences, along with accompanying sequence and BLASTN nr match information. These results are presented in Section 5.5. Note that the difference in the 'Strain-Specific Sequence' and the 'Full Sequence' can be as little as one nucleotide due to the 55 overlapping nucleotides between the strain-independent and strain-specific sequences.

Num. of Isolates with Sequence	Isolate IDs	Direction of Sequence	Side Strain-Independent Sequence is On	Strain-Specific Sequence	Full Sequence	Relevant BLAST Matches
8	MPLM35, MPLM111.2, MPLM63, MPLM15, MPLM11, MPLM9, MPLM5.1, MPLM90.1	5'-3'	3'	TTCTTCTTTGGTTTCACCACATTTAGCTGCTACAAAGGGAATTGAAGCCAATGAAGCTACTGATCCAA	TTCTTCTTTGGTTTCACCACATTTAGCTGCTACAAAGGGAATTGAAGCCAATGAAGCTACTGATCCAAG	*M. bovis* variable surface lipoproteins
7	MPLM111.2, MPLM114.2, MPLM63, MPLM15, MPLM19.1, MPLM57, MPLM90.1	5'-3'	5'	ATCTTTAGCTTCAATTCCCTTTGTAGCAGCTAAATGTGGTGAAACCAAAGAAGAAAAGAAAC	CTTGGATCAGTAGCATCTTTAGCTTCAATTCCCTTTGTAGCAGCTAAATGTGGTGAAACCAAAGAAGAAAAGAAAC	*M. bovis* variable surface lipoproteins
6	MPLM93.5, MPLM15, MPLM11, MPLM19.1, MPLM90.1, MPLM9	3'-5'	5' and 3'	CCCACAATATTAGAAACAGTTTTCATATCATAGCCTGCAATTAGCCTTTCAAGAATTT	TTGTTTCAATTGTAATTCTGTCAAATAAGTTATAAATTGATCTTGATTTAAACCCCACAATATTAGAAACAGTTTTCATATCATAGCCTGCAATTAGCCTTTCAAGAATTTCTTTAAGTCTCAACGGGCTAATGTTATTTTTATTGGTTTTG	*M. bovis* HAD-superfamily hydrolase and ISMbov-2a

5	MPLM9, MPLM15, MPLM90.1, MPLM29.1, MPLM5.1	3'-5'	3'	ATATTCATTAA CAAAGCAAAAA GCACCACAAGT TAACTTGCGGC GCTTTTTGTTG CC	CTTTTATGTTTCAGCAGGAATTCA GTCAGAAATGAAAGAACCAAAAAA GACACCAATGGCACTGTTTTTAGG GCTTTCACTTACAACTTTAATTTAC TTAATTATCGCTATTTCAATGTCGA TTAATGGTGGCTCATTTTCAAAAA TGCATGCATATTCCATTAATTTAAT GGGTCTTAAGGCAACTAATATTAT TTTTGGAATAATGAACATTTTTATT GCAATTGGAGTTTTAGGAATTGCA AATGGCTTTGCAATGTGAATGCCT AGGTATATTGAAGACTTATTGATA AAAGGTGACCTACCTTTTTGAGAA AAACTTGCTCCAAAGGTTAATCCAC GTAAACCAGTAGTAGGAATTATTT ACTCGTCAATTATTTCTGTTCCGCT AATTATCATCTTTACATTAATAGGT GCATTAGGATACATTGATACTTCA AGCTATGGGACTGTTTATGACAGT ACAATCTCTATGGCCAAGTTATATT CATTTGCTGACTTAATGGCAAATT GAAATGCCTTAGGATGCTTCTTAT TAGTTGCTTTAGCTATCTATGGTG GAATAAGAAATAGAAAAACTAATA AAGTAGAAATACCAAGTAAGAAAA AATACTTTTTGCCTACTGCATGAAT ATCAATTTTATTTGTTGCATCTGCT ATATTTATTAGTATACTTGTGCCAA TAATAAACTTATTTTTATTAATTGG CATTGACCGATCGTTAATTTCAAAT GTAGAGTTTACACAACTTTTAGTA GGCAGATTAATGCTGATTGTTGTA TTAATTCTATTTGTAAGTTTATCAA TTCTTCCAACAATTATTTTGAACAA AATCAGAACTAAAAAATTTGGTTC TATAGACAAGTACTATGAATACAC CGAAGAAAAACTAAAATCACTAAA AACACGCTTAAGCACTATAAATTA GCCTTAGAAGTTTATTTTATCTTAT ATATTAATCTATTTTGGAGCTTCTA TATGAACTTTGAAGATATATTTATT TGCACAACTGATACAGTTACTGGC ATAGGTGGGCCTGTTAATGAAAAT ACGCTAAAGTGTATTTATTACCTAA AAAATAGACCTATTAGTAAAAAAA TAATAATTTTAGTTGGGTCAATAG AACAAGCTAGAGCATTTAAAGAGT GAAATAATGAAGCTGATGACTTTG CAGCCAAATATTGACCTGGTGCAT ATTCAATAATTGTTAATGGTCAAG GTTTTAGAATGCCCAATAATAGCC AGTTGTGCCAGTTTTTACTAAAAA ATGGACCAATGTATGTTACTAGCG CTAACATTTCTGGTCAAGATCCAAT CGATATATCAGAAGCTAATAAATA CTTTCCATTAGTTAAAAATGTTTAT GACTTTGGCAGAGGTAATAATAAA GCTAGTTTTATATATAACATTGAT GAAAAAAAATGGATAAGATAACTA	*M. bovis* HAD-superfamily hydrolase and ISM-bov1

5	MPLM19.1, MPLM39, MPLM11, MPLM29.1, MPLM5.1	5'-3'	5'	AAGCAAGCATA GGTCAGTTGAA TAACTAGTATT TTTGTCATTGC TAATCATATAT AATGTAA	GATTTGACTTGGTCAATATTTCATC TCAATTCGTTTCATCAATTACATAA TGTACATAATTAAACTTGTACTTGC CGTGCACATTTAATTTCTTTTTAAG CAAGCATAGGTCAGTTGAATAACT AGTATTTTTGTCATTGCTAATCATA TATAATGTAA	*M. bovis* HAD-superfamily hydrolase and ISMbov-2a transposase
5	MPLM96.4, MPLM19.1, MPLM93.5, MPLM11, MPLM5.1	5'-3'	5' and 3'	GGGTTTAAATC AAGATCAATTT ATAACTTATTT GACAGAATTAC AATTGAAACAA AGC	TTCTTGAAAGGCTAATTGCAGGCT ATGATATGAAAACTGTTTCTAATA TTGTGGGGTTTAAATCAAGATCAA TTTATAACTTATTTGACAGAATTAC AATTGAAACAAAGCGTGGCTATGC TAAATACCAAAAGAAATGTAAATT GTGTGGTATGGACTTAAAAGGCAA AACAATATATGTAATTAGTGCTCTC AAATTTTTGAATTTAATTGAAACA AGACATGATTCAAAAAGATTAGAA AACAATTCTAAATTGCTTATGAAAT ACATTGACATTTTTAAGTTTTACAA ACAGTTACTGTACTTTTATTTG	*M. bovis* HAD-superfamily hydrolase and ISMbov-2a transposase

5	MPLM15, MPLM111.2, MPLM90.1, MPLM9, MPLM5.1	5'-3'	5'	TCAGCTTCTGA ATCTTCGCCTT CTCTTTCTGCT TCGTCGCTTTC TTGTTTAGACT CTTCTAGATCT TCTTTATCAAA TTTTTCTTCTT CTACCTTATTT TCTTCATCTTC ATAATCTTCAT TTCCTTTCTCA TCATCCTTACC TCAAGGATAAT TTTTATCTCAA GTAATACCACG ATCACGGTTTA CCGAACCTTCA TCAGGATTTCT CTCAGGCGTTT TATTTTCCCTA GGTGTTTTGTT TCCCTCAGGTG TTTTGTTTCCC TCAGGTGTTTT ATTTTCCCCAG GGTTTTTATCT CCACCTGGATT TTTGTCGGGCT CTGTGTTTTCA CCTGGGTTTTT ATCTCCTCCAG GGTTTTTATCT CCACCTGGATT TTTG	CTTTATCTTTTTTGCTTGATTCAGT GTCTGATCCTTTTTCTGATTCAGCT TCTGAATCTTCGCCTTCTCTTTCTG CTTCGTCGCTTTCTTGTTTAGACTC TTCTAGATCTTCTTTATCAAATTTT TCTTCTTCTACCTTATTTTCTTCAT CTTCATAATCTTCATTTCCTTTCTC ATCATCCTTACCTCAAGGATAATTT TTATCTCAAGTAATACCACGATCAC GGTTTACCGAACCTTCATCAGGAT TTCTCTCAGGCGTTTTATTTTCCCT AGGTGTTTTGTTTCCCTCAGGTGT TTTGTTTCCCTCAGGTGTTTTATTT TCCCCAGGGTTTTTATCTCCACCTG GATTTTTGTCGGGCTCTGTGTTTT CACCTGGGTTTTTATCTCCTCCAGG GTTTTTATCTCCACCTGGATTTTTG	*M. bovis* variable surface lipoproteins
5	MPLM39, MJ131, MPLM57, MPLM9, MPLM5.1	3'-5'	3'	AGGCCATTTTC TTGTCAGAACC ACCAAAAATTG GTATTATTTTT CCAATACTCAA ATTAG	AAAGTATCTTTTCTTTATTCTTTTT TTATGCTTTAGGCCATTTTCTTGTC AGAACCACCAAAAATTGGTATTAT TTTTCCAATACTCAAATTAG	*M. bovis* HAD-superfamily hydrolase and ISMbov-2a transposase
4	MPLM63, MPLM114.2, MPLM90.1, MPLM5.1	5'-3'	5'	TTTTAAGGCTA TAAATAGCCTA AAAATGCTTTA TAAATCTATCA ATAAATATCAA AAG	TTTGATTTTTTCATAAATTTATCCT CTCCTTTTAAGGCTATAAATAGCCT AAAAATGCTTTATAAATCTATCAAT AAATATCAAAAG	*M. bovis* variable surface lipoproteins

4	MPLM63, MPLM60, MPLM19.1, MPLM5.1	3'-5'	3'	TATTAAAAAAG TTTTAGTGTAA TGCTATATTTC TAACAATTCCG CAAGGTTTTGA ACA	TATTAAAAAAGTTTTAGTGTAATG CTATATTTCTAACAATTCCGCAAGG TTTTGAACATATTTCAACACCTTGC GTTTTTTTCATTCTGTATCATTTAT TACTTCA	*M. bovis* HAD-superfamily hydrolase and deoxyribodipyrimidine photolyase (uvrC), transposase gene
4	MPLM35, MPLM9, MPLM90.1, MPLM5.1	5'-3'	5'	CTGATTTTGCT TCAATCTCAAA TTTAAGGGTAA CTTCACCTTCG AAATCTTTAGC AGAAACAGTTA CGCTTTT	CTTTTCCTAAATCATTACCAGTAAT AGATTTTAATGTTTCGCTAAGCTTT GGTTTTGCTGCTGATTTTGCTTCA ATCTCAAATTTAAGGGTAACTTCAC CTTCGAAATCTTTAGCAGAAACAG TTACGCTTTT	*M. bovis* variable surface lipoproteins
4	MPLM60, MPLM90.1, MPLM11, MJ131	5'-3'	3'	CTTATACATTA AAAAATTCCCA ACTTTGGACAC TATATTTTTAA AATAGACTTCA ATTCTTT	CTTATACATTAAAAAATTCCCAACT TTGGACACTATATTTTTAAAATAG ACTTCAATTCTTTAATATATTTTAA TGTTTCAACATCATCTCATTCTTCT CTTTTTCTAGACTTTCTCTTCTTGA TTATTTTCGATAAATATCTAAAAGT TTCACTAATTGTTCTATTGTCAAGA ACGCCACTATTTA	*M. bovis* putative lipoprotein protein, ISMbov-2a, ISMbov-2b, and ISMbov-3a
4	MPLM9, MPLM96.4, MPLM90.1, MPLM5.1	3'-5'	5'	ATATAAATTCT TAAATTAAAAA GCACCTAATTT GAGTATTGGAA AAATAATACCA A	ATATAAATTCTTAAATTAAAAAGC ACCTAATTTGAGTATTGGAAAAAT AATACCAATTTTTGGTGGTTCTGA CAAGA	*M. bovis* ISMbov-1a, ISMbov-2a, and HAD-superfamily hydrolase
4	MPLM39, MPLM111.2, MPLM29.1, MPLM90.1	3'-5'	5'	ATATTCATTAA CAAAGCAAAAA GCACCACAAGT TAACTTGCGGC GCTTTTTGTTG C	ATATTCATTAACAAAGCAAAAAGC ACCACAAGTTAACTTGCGGCGCTT TTTGTTGCCTTTTGTTAAAATTTGA GTATCTAAATTAAAAAAAGAAAGG C	*M. bovis* HAD-superfamily hydrolase, and ISMbov1
4	MPLM63, MPLM60, MPLM90.1, MPLM11	3'-5'	5'	AGAAAAAGACA ATAGGCAAACT TATTACTGAAC GTCCAGAATCA GTTAATTTAAG G	AGAAAAAGACAATAGGCAAACTTA TTACTGAACGTCCAGAATCAGTTA ATTTAAGGCTAAATGACAATGATT ATGAAATGGACACTGTAATTGGTTT T	*M. bovis* HAD-superfamily hydrolase, and ISMbov-2a
4	MPLM15, MPLM19.1, MPLM111.2, MPLM90.1	3'-5'	3'	GCAACAAAAAG CGCCGCAAGTT AACTTGTGGTG CTTTTTGCTTT GTTAATGAATA AA	ATACTCAAATTTTAACAAAAGGCA ACAAAAAGCGCCGCAAGTTAACTT GTGGTGCTTTTTGCTTTGTTAATG AATAAA	*M. bovis* variable surface lipoproteins

4	MPLM15, MPLM37.1, MPLM9, MPLM5.1	5'-3'	5' and 3'	CAACTTGTACT TTTCCTAAATC ATTACCAGTAA TAGATTTTAAT GTTTCGCTAAG CTTTGGTTT	TGCTTTTATCTTGTTCAGCAACTTG TACTTTTCCTAAATCATTACCAGTA ATAGATTTTAATGTTTCGCTAAGC TTTGGTTTTGCTGCTGATTTTGCTT CAATCTCAAATTTAAGGGTAACTTC ACCTTCGAAATCTTTAGC	*M. bovis* variable surface lipoproteins
4	MPLM39, MPLM90.1, MPLM9, MPLM5.1	3'-5'	3'	AAAACACAGAG CCAGGCAAAAA TCCAAGTGAAA ACACAGAGCCA GGCAAAAATCC AAG	GAAAACACAGAGCCAGGCAAAAAT CCAAGTGAAAACACAGAGCCAGGC AAAAATCCAAG	*M. bovis* variable surface lipoproteins

Table A.18: The Stifle-Specific-Tropism Sequences, along with accompanying sequence and BLASTN nr match information. These results are presented in Section 5.5. Note that the difference in the 'Strain-Specific Sequence' and the 'Full Sequence' can be as little as one nucleotide due to the 55 overlapping nucleotides between the strain-independent and strain-specific sequences.

Num. of Isolates with Sequence	Isolate IDs	Direction of Sequence	Side Strain-Independent Sequence is On	Strain-Specific Sequence	Full Sequence	Relevant BLAST Matches
4	MPLM38.1, MPLM109.5, MPLM18.1, MPLM46.1	3'-5'	5'	TAAACTAGGAC AATAAAAATAG ACATTAAATTT TTATTTTTTAC AAAACTCCACT CACAATTCATG TGGAGTTTTGT AA	TAAACTAGGACAATAAAAATAGAC ATTAAATTTTTATTTTTTACAAAAC TCCACTCACAATTCATGTGGAGTTT TGTAAT	*M. bovis* HAD-superfamily hydrolase and variable surface lipoproteins
3	MPLM38.1, MJ126, MPLM61	5'-3'	5' and 3'	ATCTTTTTGTT CTTTTTCTTTA CTCTTATCTTC TTGATCGTTTT GATTACCTTGA GTTTGTGGGG	TAGTATTTGAAGAAGAATTACCAG AATCACTAGGATTATCTTTTTGTTC TTTTTCTTTACTCTTATCTTCTTGA TCGTTTTGATTACCTTGAGTTTGT GGGGAAGTATTTGAAGAAGAACTA TCAGAACCACTAGGGTCATCTTTTT GTTCTTTTTCTTTACTCTTATCTTC TTGATCGTTTTGATTA	*M. bovis* genomes with no gene-specific match
3	MPLM38.1, MPLM18.1, MPLM103.4	3'-5'	3'	AGTAAAAACTT TGATTTTTTCA TAAATTTATCC TCTCCTTTTAA GGCTATAAATA GCCTAAAAA	GCTACTGATCCAAGTAGTAAAAAC TTTGATTTTTTCATAAATTTATCCT CTCCTTTTAAGGCTATAAATAGCCT AAAAA	*M. bovis* variable surface lipoproteins
3	MPLM61, MPLM38.1, MPLM18.1	3'-5'	3'	ATAAAAAAAGA ATAAAGAAAAG ATACTTTTGAG TACTCAAAAGT GCAAAATTTTG TG	AATGGCCTAAAGCATAAAAAAAGA ATAAAGAAAAGATACTTTTGAGTA CTCAAAAGTGCAAAATTTTGTG	*M. bovis* HAD-superfamily hydrolase, ISMbov-2a, and ISMbov1

3	MPLM61, MPLM55, MPLM103.4	3'-5'	5'	CATTGGCTTCA ATTCCCTTTGT AGCAGCTAAAT GTGGTGAAACC AAAGAAGAAAA GAAAC	CATTGGCTTCAATTCCCTTTGTAGC AGCTAAATGTGGTGAAACCAAAGA AGAAAAGAAACTAGAAGCGGACAA ACCAGATCAAAGTACACCAGCTAA CCCAGATCAAGGTACACCAGCTAA CCCAG	*M. bovis* variable surface lipoproteins
3	MJ126, MPLM26.1, MPLM36	5'-3'	3'	ACTACTTTTGA TATTTATTGAT AGATTTATAAA GCATTTTTAGG CTATTTATAGC CTT	ACTACTTTTGATATTTATTGATAGA TTTATAAAGCATTTTTAGGCTATTT ATAGCCTTA	*M. bovis* variable surface lipoproteins

3	MPLM115.1, MPLM61, MPLM103.4	3'-5'	5'	AACAAAAAGCG CCGCAAGTTAA CTTGTGGTGCT TTTTGCTTTGT TAATGAATATA AAA	AACAAAAAGCGCCGCAAGTTAACT TGTGGTGCTTTTTGCTTTGTTAAT GAATATAAAATACTAAAAAATAAA AAAATTTTTAGTTCCAATTATAAGA AAATTAAAATGATAAAATAAGAAA ATAAGCATACAAGTTACTGAAAGG ATACTTTATGAAATTATTTGAATCA ATATCATCAAAAAGAGATAATAAT ATGTTGCTTAAGGCAACATTTGCT GATGAAACTAAATGCAATTTTTTA ATTAAAAAAGACAACAGCATAACT GAATATCATGAAAAGAATGAAGCA TTAGTATATTTTAGTAAGAAAGAA TCATTATCATTTTCTGACTTAGAAG GCTTTTTTAAAGGCTTAGCAGTAA ATGCCAACAGAAATTATCAAGTTG ATTTAGCTTCGTTTGCAACTGAAA AAGTTGAAATAGTTAAAGTTATTG ATGCATTTGTTAGAGCAGTTTATT TTGCAAAGGGCGAAATATTTTCAG CTAGAAAAAAAGATGAAAAGGAAG AAATTGAATTAGTTCCATTTATTGA AACTATTTCTGAACAAGCTAACGC ACAATTTAATAAATCGCTTATCTTA GCCAAAGCAACAAATTTTGCTCGT GATTTACAAATTATGCCCCCAAATA TTTGCAACTCTGAATTTTTAGCTCA AAAAGTTGCTGAAGATTTAGAACA ATACAAAAACTTGAAAGTTACTGT TTTAAAGAAAAAAGAAATCGAAGA GTTGAAGATGGGTCTTTTACTTTC AGTAAACAAAGGAAGTGTTTATGA ACCTAGAGTTGTTGTTATTGAATA CAATGGGGACAAAGATTCAAGTGA AAAGACTGTAATGATTGGTAAAGG TATTACTTTTGATTCAGGTGGATA CTCATTAAAACCTTCTAGATCAATG GTTTCAATGAAATTTGATATGTCT GGTTCAGCTATTGTTGCTGCTACA ATGAAAGCTATTGCACAATTAAAA CCAAAGAAAAATGTTTCTGCAATA ATGTGCATTACTGATAACAGAGTT AACGGTGATGCTTCACTTCCTGAT TCAGTATGGGTAGCTATGAATGGC AAAAGTGTTGAAATTAATAATACT GATGCTGAAGGAAGATTGGTTATG GCTGATGGCTTAGTTTACGGAGCA AAAGTGTTGAATGCCACTAGATTA ATTGACGTTGCAACTTTAACTGGT GCTATGGTTGTTGCACTTGGACAG ACATACACAGGCACATGGGCAACT AGTGATAAAGCTTGAGAAGACATA AAGAAAGCAGCTGAAAATGCTAAC GAATTAGTTTGAAGAATGCCACTT GATAAAGCATTTGCAAAAAACATA AAATCTTCAAAAGTAGCCGATTTA AAGAATACTGACTTTTCAGGAAAT GCAGGCTCATGTTCAGCAGCAATG TTTTTAGAAGAATTTACAGAAGGT	*M. bovis* HAD-superfamily hydrolase, and ISM-bov1

3	MPLM115.1, MPLM61, MPLM46.1	3'-5'	3'	TTTTATAAATT CTTAAATTAAA AAGCACCTAAT TTGAGTATTGG AAAAATAATAC CAA	TGTTGAAAAAGAATTTTGAGGTGG TGTTTGTTTAAATATTGGATGTAT TCCTACCAAAGCAATGCTTAGATCA ACACATGCATTAGAAGAAGTTATT CATGCGGCTAAATTTGGTGTTGTT GCTAATTTAGAAGATCTAAAAATT GACTATCAACAATCATGAGTTAAA ATGCATGAACGTAAAGCCAAAGTT GTTGCAAAGCTTTCTGGCGGCGTT AAGTTCTTAATGAAGGCATCAAAG GTACAGACTGAAGAAGGCGTTGCA AAGTTTGTTGGTGCTAGAGAAATA GAAGTTAATGGCAAGGTTTACCGT GGCAAAAATGTTATTTTAGCTACC GGTAGCCACGCTAACAGAATGAAA TTCCTTGAAGGTTTTGAAAAGGGA TACGAAAGTGGCAAGTTAATGACT TCACGTGAAGCTATTAACAATGAC AAATCATTGCCTGAATCAATGGTT ATTGTTGGTGGTGGCGTAATTGGT GTTGAATTTGCTCAAATGTATGCA TCAATGGGCACCAAAGTTACAATT ATCCAAAGAGAAGACCGTTTACTT CCTGGAATAGACAAGGAAATTGTT GACGAATTTGCTAAAATTCTTAAA ACTGAATCAAAAATTGAAGTTATC TATGGCGCAACAAGTACAAAATTA GAAGGTGACGAAAACTTAATTTAC ACCAAAGATGGCAAGGAAGAAAAA ATTACTGCTGAAGTTATTCTTATTG CTACAGGTAGAGTTCCTGCATCTG AAGGATTGGCTGAAGTTGGCATTG AATTAGGTGCTAGACGCGAAGTTA AAGTTGATAAATTCTTACGTACTA ATGTAAAAGGTGTATATGCAATTG GTGATGTTACAAACCAAAATATGT TAGCTCATGTTGCTTACATTCACGC TGTTACAGCTGTGCACCACATTTTA GATTTATATGGAATTCCATATGAT TCAACTACAAAACCAGTGCCTGCAT GTATTTACACAAGCCCTGAAATTG CTACAGTTGGTTTAACTGAAGAAC AAGCTAAAGAACAAGGATTGGACT TTTTTGTATCTAAATACAAGTTTGC AACCTTAGGTAAAGCGATTGCTGC TGAAGATACCAAGGGATTAGTAAA ATTAATTGTTCTTAAGGACGGACA CATTGTTGGTGCTTCATTAATGGG GCCTAATGTAACAGATTACGTAGC TGAATTAGCTTTAGCTATCGAAAA GAGAATTTGCGTAACTGCATTAAC TCACGTAATTCACCCACACCCAACA TTTAATGAAATTATTTGAGAAGCA GCTAGAAGTGCTTTATCAAAATTA ACTGCTGAAAAATTAAACGAAAGA AAAAACAACTAATAATTTGTTTTTA TGCTAAGCCCTGCTTAGCATTTTTT ATTTTTGCTATATGTGCATCAACAA AAATATAGGCTCAAAAAAGCGCGA TTATGACGCAAAATTATAGTTTTAT	*M. bovis* HAD-superfamily hydrolase, ISMbov-2a, and ISMbov1

3	MPLM112.2, MPLM61, MPLM18.1	3'-5'	5' and 3'	ATGGCAACTGATGAAAATGGGATACCGTTACACTACAAAATATTTCCAGGAAATGTTA	ATGGAAAATTTAAAGAAGACCAGATTGTTATAGGTATGGCAACTGATGAAAATGGGATACCGTTACACTACAAAATATTTCCAGGAAATGTTACTGATTCAAATACTTTCA	*M. bovis* putative lipoprotein protein, ISMbov-2a, ISMbov-2b, and ISMbov-3a

Appendix B

Supplemental Files

Table B.1: The scripts used in the SepSIS pipeline. The scripts and a full description for each are available at "`https://github.com/MatthewWaldner/sepsis`".

File Name	Brief Summary
`AddSampleNameToReads.py`	Adds a given identifier to the beginning of reads. For use with the SYNTH RUNMODE of SepSIS.
`CreateBamFile.py`	Creates the .BAM file needed for the SYNTH RUNMODE when given input from SPAdes and `AddSampleNameToReads.py`.
`make_fasta_from_fastg.py`	A file containing a single function that converts .FASTG formatted sequences to .FASTA formatted sequences. This file is used in the Recycler algorithm and was taken from that package [41].
`SepSIS.py`	The main file the users interacts with. It calls functions from `utils.py` and `recycle_utils.py`.
`utils.py`	This file contains the original functions and algorithms written for SepSIS.
`recycle_utils.py`	This file contains short simple functions taken from Recycler and called by `SepSIS.py`.